AR-001-985

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

ELECTRONICS RESEARCH LABORATORY

TECHNICAL REPORT

ERL-0136-TR

THE ADAPTATION AND INSTALLATION OF THE RESOURCE ACCESS CONTROL FACILITY(RACF)

J.L. Roughan and J.C. Gwatking

SUMMARY

The Resource Access Control Facility (RACF) is a software package designed to control access by users to a computer system and to data stored on the system. This report describes the modifications and additions to the functions of RACF which were made during its installation in the computing centre at the Defence Research Centre. RACF is described in sufficient detail to allow the operation of the modifications to be clearly explained. The report also summarizes the functions and standards of the computing centre and lists the actions taken to accommodate users with non-standard requirements.

Approved for Public Release

POSTAL ADDRESS: Chief Superintendent, Electronics Research Laboratory,
Box 2151, GPO, Adelaide, South Australia, 5001.

TABLE OF CONTENTS

Page No.

LIST OF APPENDICES

1. ENVIRONMENT OF THE COMPUTING CENTRE

This report discusses the installation of the Resource Access Control Facility (RACF), an IBM program product, in the central computer of the Defence Research Centre, Salisbury, South Australia. RACF is designed to control access to resources of the computer system such as data stored in the system(ref.1,2).

The Defence Research Centre, Salisbury, is engaged on a wide variety of scientific and engineering research and development for the defence forces of Australia. Substantial computing resources are required in this work for activities such as simulation, scientific data processing and engineering design. The central computer of the Defence Research Centre supplies a general computing service to the Centre. Many computer terminals are connected to the central computer via a private physically secure network in a secure area. Some terminals are installed at remote sites and connected via the telephone service but the data transmission is encrypted. Nearly all the users of the computer are cleared to access classified material but are only permitted access to material for which they have established a "need-to-know". Groups of users within the Defence Research Centre have separate interests and are administered separately. In all there are about 500 active users, most of whom are engaged on scientific and engineering projects; very few have substantial formal training in computer programming.

Data owned by users of the computer system are stored on disks attached to the system and on magnetic tapes which are stored in a physically secure area adjacent to the computer room. Many disk data sets are archived to magnetic tape to provide adequate free space on the disks. The archival is regular and automatic but commands are provided so that users can easily retrieve and manipulate archived data sets. The archives contain many more data sets than can be stored on the disks(ref.3,4,5).

Various software packages such as IMS, STAIRS and GIS allow users of the system to access data bases. TSO, a time sharing system, is used by a majority of the users to enter and edit data and programs, test new programs, and run programs and inspect their output.

A number of requirements must be met by a security package such as RACF in this environment. Most of the actions of the security system should be automatic and transparent to the user, to reduce inconvenience and to lower the possibility of user error. In particular all data stored on the computer, whether on disk, on magnetic tape or in the archives should be automatically protected as it is created. While access to this data should be restricted to its owner as a default, it should be relatively easy for the owner to share his data with other users on a need-to-know basis. It should be possible to specify that some users may only read the data while others may both read and alter it. It is important that the sharing of data should be readily controlled by the owners of the data and not by a central administrator. In order to control access to data and provide reliable privacy and integrity the security system must be able to identify users and monitor their activities. It should provide data to enable the production of user and management reports describing these activities. The overhead of providing protection for a large amount of data should be low and the security software must be properly integrated with the normal computer software so that full integrity of the system is maintained.

The basic RACF product fulfills most of these requirements for the control of access to data. This report describes the work which has been done in this computing centre to obtain these facilities using RACF. Appendix II describes the use of RACF procedures for users of the computer system.

2. HISTORY OF THE COMPUTING CENTRE

The table below gives a brief history of the size of the computer system(ref.6,7,8).

Year	Machine	Online storage	Terminals	Users
1961-1976	IBM 7090	-	-	200
1975	IBM 370/168	800 MB	15	7090(150),168(100)
1976	IBM 370/168	1400 MB	40	7090(50),168(250)
1977	IBM 370/168	1600 MB	60	300
1978	IBM 370/168	4000 MB	80	350
1979	IBM 370/3033	6000 MB	110	400

The growth and diversity of the user population and the increase in online storage and the number of interactive terminals implied a need for a comprehensive means of controlling the use of data. The technique used before the installation of RACF was password protection. Password protection had the disadvantages that it was not automatic and it was cumbersome to use because each protected data set required a password. Security exposures were possible since all the users who needed to use a data set had to know the password to that data set. It was difficult to regularly change passwords because of the difficulty of informing all the users of the data.

The installation of RACF has overcome these problems and provided a very flexible and powerful method of controlling the access to data.

3. CURRENT SECURITY MEASURES IN THE COMPUTING CENTRE

3.1 Physical security measures

The building housing the computing centre is located in a secure area to which entry is controlled by an identifying pass. Access to the computer room is further restricted. The main communications' network is private and physically located within the secure area. Links to terminals in other secure areas are encrypted because they use the public telephone network.

Adequate fire detection and prevention equipment is installed in the computer room and tape storage area.

Backup tapes of all the disks and duplicate copies of all archive tapes are kept in a separate building. (The archives contain disk data sets which have been transferred to tape to provide adequate unused disk space).

3.2 Procedural security measures

Nearly all the users of the system are cleared for access to classified material and owners of data may allow such users to access their data on a 'need-to-know' basis. Users of the system not cleared for access to classified material are restricted by the security software (RACF) to their own data plus essential system data even if another user tries to allow them access to his data.

All disk data sets which are changed during a day are backed up on to tape during the evening(ref.9). Data sets can readily be restored from the backup tapes or entire disks can be reconstructed. The operational housekeeping procedures are designed so that recovery of data is always possible(ref.10).

3.3 Software security measures

A record of accesses by users to data sets is maintained and reports are distributed to users each fortnight showing which other users accessed their data sets and what the types of access were (e.g. read or write).

Each user is personally identified when running a batch job or when logging on to a time sharing terminal by a user identification and password. Each user's password is known only to the user and can be changed by the user at any time; users are in any case forced to change their passwords every three months.

RACF controls the access to all data on the basis of information provided by the individual owners of the data.

4. FUNCTIONS PROVIDED BY RACF

4.1 RACF functions

RACF conveniently supports most of the requirements of this computing centre for controlling access to data. In addition, RACF has been designed so that it is easy to modify or extend its functions. Consequently the work described in this report was undertaken to extend RACF to provide the extra functions required in this computing centre.

There are certain fundamental requirements of a security system and RACF meets these:

(a) the security system must be fully supported by and integrated with the operating system of the computer,

(b) there must be no loopholes or exposures by which the access control may be bypassed except by a hardware failure; even this should not result in a general exposure,

(c) it must be possible for all data to be protected automatically - without specific user action, and without severe overhead,

(d) it must be possible for the owners of data to control the access to their data,

(e) reliable reports of successful and failed accesses to data must be available to the owners of the data,

(f) users of the system must be reliably identified so that access to the system can be controlled,

(g) it must be possible to limit the type of access to data to input only or to input and output,

(h) it must be possible to control the access to data by different users independently.

RACF allows individual users, groups of users, or all users to be given access to a data set. Different users or groups of users may be allowed different types of access (e.g. input only or input and output).

The types of access controlled by RACF are READ, UPDATE, CONTROL, and ALTER. READ allows input while UPDATE allows input and output. CONTROL is of specialised interest and may be considered as being equivalent to UPDATE. ALTER allows input, output and deletion of data sets.

4.2 Major problems in this computing centre

There are several major areas where the standard functions of RACF do not meet the requirements of this computing centre. These problems have been solved by a variety of techniques, including modifying and extending

RACF. The solutions are discussed in Section 7.

(a) A RACF definition or profile must exist in the RACF data set for every protected disk data set. In our case, since our archives are in effect an extension of the disks, a profile would also need to exist for every data set in the archives. The RACF data set would in our case become rather large and access to a profile would involve greater overhead.

When data sets are created it is possible to create the RACF profile automatically by copying some model profile. However if the user alters his model, the previously created data set profiles will not change.

(b) The use of RACF to control access to data sets stored on magnetic tape is very awkward. No provision is made for erasing tapes when data sets are deleted.

(c) In this computing centre, access to data sets in the archives must be controlled by RACF in a similar way to the control of access to disk data sets.

(d) Disk space which has been freed by deleting, moving or compressing data sets can easily be used for input by any user, without first writing on the space. This is a major privacy exposure.

(e) In this computing centre, operator started tasks are used extensively to submit batch jobs to perform operational housekeeping functions on the computer system. These submitted jobs require passwords on the job cards but it is not possible for a started task to obtain the password corresponding to its userid.

(f) RACF does not provide any means for printing the security classification on printed output. The RACF LEVEL parameter could perhaps be used to maintain the security classification of a data set but it would require major changes to the JES2 job entry system software to cause automatic printing of the classification on a job output.

4.3 Minor problems in this computing centre

Many other problems or inadequacies exist in the operation of RACF which do not have a major impact. Most of these problems, which are described below, have been solved by the work described in Section 8 of this report. The problems as yet unsolved do not cause any security exposures but cause minor inconveniences for the users.

(a) FORTRAN programs open a data set for INOUT processing even when only READ statements appear in the program. READ access authority is therefore not sufficient to be able to use a data set for input to a FORTRAN program. At least UPDATE access authority is required.

(b) The access available to each generation of a generation data group (GDG) must be defined to RACF when the generation is created. This could become tedious and induce errors with the frequent creation of new generations.

(c) No provision has been made in RACF for an access authority of execute only, which should be a more restricted access type than READ. Presumably it would be difficult to make the MVS operating system properly support such a RACF access level.

(d) The user's password on a batch job is checked at the time the job begins execution, not at the time the job is submitted. Thus, if the password is changed by the user after a job is submitted but before it begins execution, then the job will fail.

The RACF password command to change the password during a time sharing session does not change the password in the TSB (a system control block). Since the TSO SUBMIT command obtains the password for submitted batch jobs from the TSB, any job submitted after the password is changed in a session will fail.

If a job card is included in the JCL (job control statements) of a batch job submitted using the TSO SUBMIT command, then the password is not inserted in the job card. However, any job cards built entirely by the SUBMIT command do include the password.

The need to include the password in a batch job submitted as a card deck is a major security exposure for the password, but this is not logically a RACF problem.

(e) The RACF manuals(ref.1,2) and the numerous RACF commands are too complicated for users who are not primarily programmers.

In issuing RACF commands to define the access available to a VSAM data set, the commands have to be issued separately for the components of the data set (cluster, index and data).

RACF only issues write-to-operator(WTO) error messages which normally would appear on the SYSLOG printout of a batch job but not at a time sharing terminal where required by most users.

(f) The use of DD DATA JCL statements is an exposure since a job could read the JCL of other jobs following it in a card reader's input stream.

(g) RACF does not allow any user to create a data set in the name of another user.

(h) Data set access statistics can be recorded in the RACF profiles of data sets. However only a count of accesses by each user in the access list is recorded. The actual level of access (rather than that allowed) or the date of access is not recorded. The count of accesses cannot be reset to zero.

5. EXPLANATION OF THE OPERATION OF RACF

This explanation should not be regarded as a complete, or even fully accurate description of RACF. Some knowledge of general IBM 370 operating system functions has been assumed (however a number of definitions appear in the glossary).

RACF stores in a special data set a record or profile for every entity or resource to which it controls access. The profile for a resource contains a description of the level of access permitted to the resource. A data set is an example of a resource.

RACF is installed as an integral part of an operating system, MVS, which controls the operation of an IBM 370 computer and provides user services such as job management and data management. The installation of RACF includes the modification of certain parts of MVS. The modifications involve the insertion of code to invoke RACF to perform three broad functions :- to check whether a user has the authority to access a resource (known as the RACHECK function), to verify the identify of a user entering the system (RACINIT) and to manipulate the profiles of protected resources (RACDEF).

For instance, RACHECK macros have been inserted in the MVS OPEN routines and in the MVS routines which delete or rename data sets. The macro is executed before any access to a disk data set which is RACF protected, as indicated by a protection flag set in the control block (DSCB) pointing to the data set in the directory of contents (VTOC) of a disk. The RACHECK macro is also executed before any access to a standard labelled magnetic tape if the RACF option for protection of tapes is enabled. Execution of the RACHECK macro causes an SVC interrupt which invokes the RACF RACHECK SVC routine. This SVC routine checks the authorization of the user to access the resource at the requested level, for example READ or UPDATE. The SVC routine returns a code to the calling routine indicating whether the user may access the resource. Messages may be issued by the SVC routine and if access is denied the routine which executed the RACHECK macro usually causes an ABEND (abnormal termination of the user program).

MVS has been modified to execute a RACINIT macro when a batch job or started task begins execution, when a time sharing user logs on or when a user logs on to the data base management system. The RACINIT macro causes the RACF RACINIT SVC routine to be invoked which checks whether the user's password is correct and sets up an MVS control block (ACEE) defining the characteristics of the user. The characteristics of the user are defined in a user's profile in the RACF data set, where his password is also stored. The security administrator may alter the user's profile. An important parameter that can be set in the profile specifies that all disk data sets created by the user are to be automatically protected by RACF: that is, the flag is to be turned on in the DSCB and a RACF profile is to be defined for each new data set.

Profiles in the RACF data set for disk data sets or tape volumes can be created, modified or deleted by executing a RACDEF macro. The RACDEF macro invokes the RACF RACDEF SVC routine to perform the required operation on the profile. RACDEF macros have been inserted in MVS routines which create, move, rename, extend or delete disk data sets so that corresponding creation, modification or deletion of the RACF profiles of the data sets will occur automatically.

Unfortunately similar provisions have not been made in the case of magnetic tape data sets. Specific action needs to be taken to create, modify or delete RACF profiles for magnetic tape volumes. Note that RACF only protects magnetic tapes by volume, not by data set, recognizing that once a data set is opened on a volume, other data sets can be accessed on that volume without repeating the open. Thus it is sensible to only protect volumes.

It is possible for "authorized programs" to use the various RACF macros to enhance the functions available from RACF. An authorized program is a program permitted to perform supervisor functions.

RACF provides commands for users to allow access by other users to their disk data sets and tape volumes. Specific users can be given access or all users can be given access. The level of access granted may be NONE, READ, UPDATE, CONTROL or ALTER. The first three are self explanatory, CONTROL is not usually required, and ALTER allows all forms of access to a data set, including the ability to delete or rename. Specific users can be given any of these levels of access and all other users can be given any one of these levels of access to a data set or magnetic tape volume.

Under RACF, users can be connected to a RACF Group. RACF Groups are designed to simplify data set creation and sharing for a project oriented group of users. A Group data set is identified by prefixing the data set name by the Group identifier, just as the owner of a user data set is identified by prefixing the data set name by the user identifier. Users connected to a RACF Group may use and optionally create Group data sets. This reduces the impact of the RACF restriction that one user may not create a data set for another.

The definition of which users are permitted to access a data set may be simplified by including a Group name in the list. Then any users connected to the Group may access the data set.

6. MODIFYING THE FUNCTIONS OF RACF

As previously mentioned, RACF incorporates several features designed to enable individual computing centres to modify or extend its function. Those features relevant to the work described in this report are explained in detail below. Other features are mentioned only briefly.

6.1 Performance

RACF contains a number of facilities to change its performance, that is to reduce overhead or to make it more efficient. Facilities for recovery are also supplied. However this description will concentrate on those facilities which allow the functional behaviour of RACF to be changed.

6.2 New resource classes

New classes of resources to be protected may be defined to RACF. This feature has not been used at this computing centre.

6.3 RACF macros

The RACF macros RACHECK, RACDEF and RACINIT execute the respective SVCs and can be used by programs written by an computing centre to add additional functions to RACF. The RACDEF and RACHECK macros are used by the archiving programs used in this computing centre since these programs bypass normal RACF processing. The RACDEF macro is also used in this computing centre to provide automatic RACF protection for a pool of magnetic tapes available to all users for the storage of large catalogued data sets. The RACHECK macro is used to authorize certain RACF commands and the creation of data sets that would normally be prohibited.

6.4 RACF exits

RACF provides flexible exit facilities to allow a computing centre to add or alter many functions. An exit is a program (subroutine), written and installed by the computing centre, which is called by RACF at a certain stage when processing a request to RACF. The exit is able to modify parameters of the request and supply a return code to cause the request to fail, to be repeated, to ignore validity checks or to terminate but return a successful completion code.

The exits supported by RACF are given access in a flexible manner to most of the parameters used in processing the respective requests.

6.4.1 RACDEF exit

The RACDEF SVC is used to define, alter or delete RACF profiles for protected resources. RACDEF is executed by MVS routines which create, alter or delete DASD data sets.

The RACDEF pre-processing exit is called by the RACDEF SVC before any RACDEF processing has occurred. The return codes from the exit may bypass normal RACDEF authorization checking, terminate RACDEF processing, or refuse authorization for the RACDEF. The main functions of the exit in this computing centre are to prevent the creation of a RACF profile for every disk data set which is created (see Section 7.1 for more details), to prevent attempts to delete RACF profiles when data sets without profiles are deleted, and to allow users to create data sets for other users who have given them ALTER authority in their default profiles.

6.4.2 RACHECK exits

The RACHECK SVC is used to check the authorization of a user to use a resource. RACHECK is executed by MVS routines such as OPEN to check whether a user is authorized to open a data set with the requested level of access.

The RACHECK pre-processing exit is called by the RACHECK SVC before any RACHECK processing occurs. The return codes from the exit may cause RACHECK to fail, allow authorization without further processing, or allow authorization but with further processing, such as logging. The main functions of the exit in this computing centre are to provide a fast path for a user's own data sets (that is provide access with no further checking), to detect disk GDGs and cause the check to be made on the GDG base name instead, and to simulate expiry date protection for all system data sets by requiring an operator authorization even when access is permitted.

The RACHECK post-processing exit is called by the RACHECK SVC after most RACHECK processing (except the issuance of error messages) has occurred. The return codes from the exit may cause the RACHECK to be repeated (including the execution of the pre-processing exit). Obviously some of the parameters for the RACHECK would have been changed by the exit before this retry. The exit may also modify the completion code to be supplied by the RACHECK SVC. The main functions of the exit in this computing centre are to issue a RACDEF to define a tape profile if one does not exist, and to retry RACHECK with a user's default data set profile for data sets which are not defined to RACF.

6.4.3 RACINIT exits

The RACINIT SVC is executed when a user accesses the computer system or at the end of a job or session. RACINIT is issued by MVS at job start and end, TSO logon and logoff or IMS logon and logoff.

The RACINIT pre-processing exit is called before much RACINIT SVC processing has occurred. The exit may set a return code to cause the RACINIT to fail or to be accepted without further RACINIT processing. The exit is mainly used in this computing centre to supply userids for batch jobs from the first three characters of the jobname and to prompt the operator for the userids of started tasks not already defined to RACF. Started tasks (that is, jobs started by operator START commands) can be defined to RACF in a table which indicates the userid and Group associated with them. The userid and Group of a started task not in the table can be entered by the operator when prompted.

The RACINIT post-processing exit is called after most RACINIT SVC processing has occurred. The exit may set a return code to cause the entire RACINIT request to be retried with parameters changed by the exit. The exit may also alter the completion code which will be returned by the RACINIT SVC routine to the program which executed the RACINIT macro. The exit is mainly used in this computing centre to request permission from the operator for special users to log on and to store the password of the user in an area of main storage. (The password can then be obtained by a job which needs to submit another job, and included on the generated JOB card).

6.4.4 RACF command exit

The RACF command pre-processing exit is called from various RACF commands before any command processing has occurred. The exit may set return codes to cause a command to fail with or without an error message, or to be accepted without any authorization checking. The exit is used in this computing centre to allow certain commands to be authorized which are normally forbidden. The commands are necessary because not all data sets have RACF profiles in this computing centre. Sections 7.1, 7.3, 8.5 and 8.6 supply more information and the Appendix gives full details.

7. EXTENSIONS TO RACF FUNCTIONS

Most of the problems described in Section 4.2 have been solved in this computing centre. This section describes the solutions, which required the development of a number of exits and TSO command procedures (CLISTs).

7.1 Default definition of access to disk data sets

To simplify the use of RACF, and to reduce the size of the RACF data set, most disk data sets are not given a RACF profile but instead are defined by a single default profile for each user or Group. Each user may easily modify his default profile so that the access available to all his data sets (except those specifically defined to RACF with profiles) may be easily altered.

RACF normally expects a profile to exist for each data set and so several RACF exits are used to allow the default profiles to be used when data sets do not have profiles of their own.

For instance, a RACHECK (to check the authorization to access a data set) may discover that no profile exists for the data set. The RACHECK post-processing exit routine detects that no profile was found and modifies the data set name to be checked to the name of the default profile of the user or Group owning the data set. The exit then returns a code causing the RACHECK to be repeated. The exit also sets a flag which can be tested by other exits indicating that no profile was found. When the RACHECK is repeated the default profile is found and used to provide the access list for the data set.

When a RACDEF macro is executed by a system module responsible for deleting, renaming, moving or extending a data set, a RACHECK is first performed by the system module to test the authorization for the action. Therefore the RACDEF pre-processing exit may test the flag set by the RACHECK post-processing exit indicating whether a profile exists for the data set. If a profile does not exist then the RACDEF pre-processing exit returns a code to cause the RACDEF to be aborted but with a successful completion code. Thus deletion etc. of the data set continues successfully without errors being caused by an attempt to delete a nonexistent RACF profile.

When a data set is created, a RACDEF is executed to create a profile for the data set (assuming that data set protection is automatic - a RACF option). In this computing centre, the RACDEF pre-processing exit returns a code to cause the RACDEF to be aborted but with a successful completion code. Thus all newly created data sets do not have a specific definition or profile in the RACF data set but the RACF protect flag is switched on in the control block (DSCB) pointing to the data set in the disk directory (VTOC).

A RACF command can be executed to create or modify a profile for a data set. That is, data sets can be specifically defined to RACF, overriding the access list defined in the default profile. A data set profile can also be deleted, thus causing the access to the data set to revert to that defined by the default profile. The versions of the RACF commands to add or delete RACF profiles without switching on or off the RACF protect flag in the DSCB are non-standard since RACF normally expects all protected data sets to have a profile. RACF normally prohibits the use of these commands except under very restricted conditions. The RACF command pre-processing exit is used to allow wider use of the above commands. The exit executes a RACHECK for the data set. If the user has ALTER authority, then the exit returns a code causing the command to be accepted without any authorization checking. (ALTER is the highest level of access authority to resources available in RACF).

A CLIST has been designed to simplify the use of the RACF commands. The CLIST executes a TSO command designed to search the catalog or archive catalog for the data set, discover the data set type and location, and issue a RACHECK to detect whether the data set has a profile or not. Then the appropriate RACF commands are built and executed by the CLIST.

Another CLIST to display the access available to data sets has been designed. The CLIST displays the default profile if a profile does not exist for the data set.

Disk Generation Data Groups (GDGs) may be defined by the GDG base name. The RACHECK pre-processing exit modifies a GDG generation name to the base name. If a profile does not exist for the base name, then the RACHECK is retried using the default profile just as for an ordinary disk data set (see Section 8.2).

7.2 Automatic protection of magnetic tape data sets

RACDEF macros are not automatically executed by MVS to create tape volume profiles during the creation of a data set on a tape volume which is not already defined to RACF. Also, tape profiles are not normally deleted when all the data sets on a volume are uncatalogued.

RACF exits and other programs are used in this computing centre to automatically define and delete tape profiles and to allow access to tape data sets to be defined by the default profile or by a specific definition just as for disk data sets. All standard labelled tape volumes containing catalogued data sets in this computing centre are defined to RACF and have profiles in the RACF data set. However a flag in the installation data of each profile is used to indicate whether the owner's default profile or the actual volume profile is to be used to define the access available to a volume.

The RACHECK post-processing exit checks the flag in a tape profile and causes the RACHECK to be retried with the owner's default profile if indicated. Otherwise the exit allows the RACHECK to complete using the actual tape profile. If no profile for the tape volume exists, then the RACHECK post-processing exit executes a RACDEF macro to create a volume profile. Note that if RACF tape protection is active, then a RACHECK macro is executed by MVS during the creation of a new data set on a standard labelled tape, thus ensuring that profiles will exist for all tapes containing data sets.

The RACDEF pre-processing exit sets the flag in the tape volume profile it is creating to indicate that the owner's default profile should be used to define the access available to the tape. Subsequent RACHECK requests therefore must be able to determine who the owner of the data set on the tape is. However the data set name is not available to the RACHECK exits as it is for a disk data set, and it would be too complicated to modify every MVS module that issues a RACHECK for a tape volume to make it available. The compromise adopted is to modify only the MVS OPEN module(Appendix VIII) that handles the creation and extension of tape data sets and to pass the data set name to the RACHECK SVC by way of an installation parameter. Whenever the RACHECK post-processing routine determines that the tape volume does not already have a profile it issues a RACDEF macro to create one, again passing the data set name. The RACDEF pre-processing exit then stores the owner, as derived from the data set name prefix, in the installation data field of the profile, thereby making it available to subsequent RACHECK requests. This technique does mean that each protected tape volume must have a profile, whether the protection is defined by the owner's default profile or not.

In this computing centre, a program is run regularly to determine which standard labelled tapes do not contain catalogued data sets. The program causes all such tapes to be erased (except for the internal volume label) and to be placed back in the scratch pool. The program which erases a tape executes a RACDEF macro to delete the RACF tape volume profile. Thus when a new data set is subsequently created on the tape, a profile defining its new owner can be created as described above.

The CLISTs (time sharing command procedures) referred to in Section 7.1 above, which modify or display the access available to disk data sets, also modify or display the access available to tape data sets in an identical manner. The CLISTs execute a TSO command which searches the catalog to discover whether the data set is stored on tape. A RACHECK is issued by the command to detect whether the flag in the installation data indicates that the actual volume profile or the default profile is used to define the access available to the tape. Then the CLISTs execute appropriate RACF commands to modify or display the tape profile or display the default profile.

From a user's viewpoint the same technique is used to define specific access to a tape data set or to cause the definition of access to revert to the default profile as for a disk dataset. However, since the same tape volume profile applies to all data sets on the volume, then altering the access available to any one of the data sets will obviously have the same effect on the others.

Tape Generation Data Groups (GDGs) cannot be treated in the same way as disk GDGs since the RACHECK exits cannot detect that a data set is a GDG - the exits have no access to the data set name for tapes. Thus if the GDG requires a different level of access from that provided by the default profile for the user or Group, then each generation must be defined specifically when created.

7.3 Protection for datasets in the archives

This computing centre operates an archiving scheme(ref.3,4,5) that removes infrequently used data sets from the disks allocated for the storage of user data. These data sets are either written to tape (there may be several hundred per tape) or are compacted and stored as part of a special data set on another disk.

Since the archives are really an extension of the disks, the data sets in them must be afforded the same protection they would have if they were still on disk. The archive tapes and the special disk data set are therefore protected by RACF against all accesses, since they contain data belonging to many users. The programs of the archiving scheme that access these resources use a special MVS feature that enables them to bypass all RACF processing. Given this privilege, the programs must ensure that the users invoking them have the necessary authority to perform the desired action on the requested data sets. To accomplish this the programs issue their own RACHECK macros. ALTER access is required to perform any operation except for the RETRIEVE or RELOAD functions, which require READ access.

When transferring data sets between the archives and disk and vice versa, or when deleting data sets from the archives or disk, the programs must also perform the appropriate operations on the profiles of those data sets that are specifically defined to RACF. The programs issue

RACDEF macros to perform this function. The RACDEF pre-processing exit allows all processing requested by any program of the archiving scheme to proceed without authorization checking.

When a data set is transferred to the archives it may or may not be deleted from disk, depending on whether the operation is ARCHIVE or BACKUP, respectively. If the data set is specifically defined to RACF then its profile is copied and the volume field in the copy changed to 'ARCHIV'. This profile then protects the copy of the data set in the archives. If the disk data set is deleted then the original profile will also be deleted. The reverse processing is performed when a data set with a specific profile is returned to disk by the RETRIEVE facility, which also deletes the copy in the archives, or the RELOAD facility, which does not. In these cases the volume field of the profile that is created for the disk data set is changed from 'ARCHIV' to the serial number of the disk volume chosen to receive the data set.

Other features of the archiving scheme, such as the deletion or renaming of data sets, similarly manipulate the profiles of those that are specifically protected.

The command procedures (CLISTs) created to define access to disk data sets and list the access available to them (see Section 7.1) also operate identically on data sets in the archives. The default profile associated with each user protects all data sets in the archives that are not specifically defined to RACF, just as it would if those data sets were still on disk.

To enable the archiving programs to issue RACDEF macros to define profiles for data sets in the archives a modification to the RACDEF SVC was necessary. These profiles specify 'ARCHIV' in the volume field. This is a fictitious volume that simply indicates that this profile applies to a data set in the archives, rather than to another copy of the data set that might exist on disk. However the RACDEF SVC rejects attempts to create profiles for data sets on volumes not currently online. This restriction has been removed by this computing centre when the volume is 'ARCHIV'. It still applies to all other volume serial numbers. The modification was made to CSECT ICHRDF00 of the module IGC0012C (see Appendix VII).

7.4 Erasing released disk space

It is easy for any user to access and read information in disk space which has been released by deleting, compressing or moving a data set.

Several solutions to this problem may be proposed:

(i) Erase all disk tracks during or subsequent to the release of the space - possibly unacceptable because of the overhead incurred by the extra channel and disk activity.

(ii) Encrypt all data which is protected by RACF against general inspection except perhaps by a specific list of users. The overhead in this computing centre would be great since all our default profiles have this characteristic - we do not allow a user to provide even READ access by all users to all his data sets.

(iii) Use the RACF Level concept to indicate which data sets need erasing and erase these during the release of the disk space. This would be unacceptable because it is likely that users would forget to set the Level.

(iv) Erase all data which is protected by RACF against general inspection. The decision would depend on the result of an RACHECK which would involve more overhead for the average sized data set than simply erasing the data set.

The most feasible solution is the first. However, even this may introduce an unacceptable increase in channel and disk activity. We have implemented this method(Appendix X) and intend to measure the consequent change in performance. The channel command used to erase each track will not cause the channel or control unit to be busy during the erasure. Only the actual disk drive will be occupied and even it will be available to other tasks between tracks.

A satisfying solution would involve a hardware addition to a disk drive which allowed a flag to be set (with low overhead) which would prevent a track from being read until it had been rewritten. If a track was only partly rewritten, the remainder of the track should be unreadable.

7.5 Accessing the password in a started task

It is useful for a program to be able to obtain the user's password so that it can build the job control statements (JCL) for another job and then submit the job so constructed for execution. (The password must appear on the JOB card of each job).

An interactive program (run using TSO at a terminal) can obtain the user's password from the TSB (an MVS system control block). In this computing centre, the RACINIT post-processing exit has been used to place the password of the user for a batch job in a region of storage accessible to the user. Thus batch programs can also obtain the user's password.

Since a password is not needed to run a started task (a job run by an operator START command), the RACINIT exits do not have access to the password. Also there is no standard way for even an authorized program to gain access to a user's password from the RACF data set.

In this computing centre a task has been set up which executes at every IPL (system initialization) and generates a random password once per day for the userid associated with operations jobs. The password and date are stored in a data set only accessible by operations jobs and a PASSWORD command (a RACF command) is issued to reset the password for the operations user.

A started task, if defined as owned by the operations user, can then obtain the password from the data set in which it is stored. It is not normally possible to log on to time sharing (TSO) with the operations userid since the password is unknown.

7.6 Printing the security classification

A modification to JES2 (a job entry subsystem of MVS) has been designed to print RESTRICTED, CONFIDENTIAL or SECRET on each printed page of a data set in SYSOUT classes R, C or S respectively. This security classification is also repeated on the separator pages.

The number of lines per page available to a user for SYSOUT classes R, C and S has been reduced from 66 to 60.

Another modification to JES2 causes the operator to be warned on the separator pages that a job contains classified output if a certain character appears in the job name. This is useful when the classification is included as part of a text data set, for example, and is not inserted on the output by JES2.

8. SOLUTIONS TO RACF PROBLEMS

This section addresses the problems described in section 4.3. Circumventions and solutions to some of the problems have been found and implemented by this computing centre, while solutions to others have not yet been implemented due to their difficulty.

8.1 FORTRAN I/O

FORTRAN programs open data sets INOUT or OUTIN, depending on whether the first statement issued for the dataset is READ or WRITE, respectively. Thus a FORTRAN program needs at least UPDATE access for all data sets, even though only READ statements may be used. This can be reduced to READ access by using the IN subparameter of the LABEL parameter on a DD job control statement, or by using the INPUT parameter of the time sharing (TSO) ATTRIB command in conjunction with the ALLOCATE command. Both these techniques are fairly awkward, particularly the latter.

In addition, the ATTRIB command cannot be used when allocating concatenated data sets under TSO, so that under these circumstances UPDATE access must be available to each of the data sets in the concatenation.

A reasonable solution to the problem would involve modifying an INOUT OPEN request to INPUT if only READ access is available to the data set. The modification could be performed by the RACHECK exits during OPEN processing and restricted to FORTRAN programs by examining the form of the DDNAME. However if the program later attempted to write to a dataset that had been only opened for INPUT the resulting diagnostic error message would not be particularly simple to understand (contrasting with the RACF error messages which are very lucid). Techniques for modifying the OPEN as suggested above are not known and grave difficulties have been predicted.

Alternatively the FORTRAN library routines that handle OPEN requests could be modified to intercept INOUT requests and, if the RACHECK denies UPDATE access to the data set, to reissue the RACHECK for READ access. If this check succeeds the routines could then modify the OPEN to INPUT and resume processing. However existing load modules would need to be relinked to incorporate the new version of the library routines.

We believe that the latter solution, although not ideal, offers the better chance of success.

8.2 GDGs

The obvious requirement is to automatically protect all generations of a GDG in the same way. This is accomplished for disk GDGs by detecting the form of a GDG data set name in the RACHECK pre-processing exit and modifying the name to the GDG base name. The RACHECK is then carried out on the GDG base. If the base has been defined specifically to RACF, then access is authorized accordingly. If not the default profile is used to determine authorization. The commands to provide access to data sets include provision for defining GDG bases to RACF and listing the profile.

Unfortunately, it is difficult to manage tape GDGs in the same way since the data set name is not available to the RACHECK exits. The result is that each generation must be specifically defined or the default profile will be used. No satisfactory solution to this has been found.

8.3 Execute only access

It is difficult to see how this could be provided given the structure of MVS. However, it is highly desirable and MVS should be modified to allow this additional level of access to be controlled by RACF. Obviously the level EXEC would fall between NONE and READ in the hierarchy of levels of access.

8.4 Passwords

(a) Add password to JOB card in SUBMIT

With the TSO Command Package (IBM program number 5740-XT6) installed, the password is inserted on JOB cards created by the TSO SUBMIT command. However if a job processed by the SUBMIT command includes a JOB card, then the SUBMIT command does not add the password to this JOB card.

A SUBMIT exit has been written by this computing centre to perform this function. The exit also changes the userid field in the jobname (the first three characters) to the userid of the submitter (the RACINIT exits allow the RACF USER parameter to be omitted and obtain the userid of a batch job from the jobname).

(b) Passwords in card decks

The need to include a password in card jobs creates a risk of compromise of the secrecy of the password. DRCS practice is for all card decks in the Centre to be stored in locked cabinets. The password must be punched using print inhibit on a JOB card continuation which is destroyed by the operator whenever the job is submitted.

(c) Checking batch job password at submission time

The password should be checked at job submission time rather than at initiation of execution, because the user could have changed it in the intervening period. The modification required is to issue a RACINIT macro in the IEFUJV SMF exit at JOB submission time to check the password on the job. The caller will be identified by an installation parameter in the macro parameter list. When the RACINIT is issued at job initiation, the RACINIT exits will bypass the need for a correct password on the job.

(d) Password changes during a session

If the password is changed using the PASSWORD command during a TSO session, the change is not reflected in the TSB (an MVS control block). The SUBMIT command obtains the password for batch jobs from the TSB and thus batch jobs would fail if submitted after the change. This problem has been circumvented at this computing centre by not supplying users with documentation on using the PASSWORD command to change passwords. Passwords are only changed at logon or in a batch job.

8.5 Simplifying the use of RACF

(a) Simplified commands

Standard RACF has over twenty rather complicated commands. In this computing centre command procedures (CLISTs) have been designed to simplify the commands which have to be used and reduce their number. Only two commands are needed by most users, and administrators of RACF Groups need to use one or two more. The two main commands define the access available to a data set and display the access available to a data set. Disk, tape and archived data sets are treated identically as far as the user is concerned. Thus the disk data set commands of RACF and the RACF commands for tape resources are amalgamated. As well as this the effect of the RACF PERMIT command is included. The ability to specifically define a data set to RACF or cause it to revert to the definition of the default implies inclusion of the effect of the RACF ADDSD and DELDSD commands.

The CLISTs execute a specially designed command which issues various macros to search the catalog and the archive catalog for the data set name. The volume and type of the data set are identified. The RACHECK macro is executed to discover the owner of the data set for tape data sets and whether the default profile is to be used. The results of this special command are passed back to the CLISTs.

If a VSAM data set is identified, the cluster, index and data entries are automatically and identically defined to RACF. (VSAM stands for Virtual Storage Access Method).

The CLISTs allow easy revision or display of the default profile, which defines the access available to all data sets not specifically defined to RACF. If the access available to a data set not specifically defined to RACF is requested, the default profile is displayed with an explanation.

The CLIST used to define access to data sets causes various RACF commands to be executed. Some forms of these commands would not be allowed by RACF but for the action of the exits described in Section 6.3.

The command exit authorizes the use of ADDSD and DELDSD commands with the NOSET parameter for any data set to which the user has ALTER authority rather than only to his own data sets. (The NOSET parameter is necessary because disk data set profiles in this computing centre must be created and deleted without affecting the RACF protect flag in the DSCB).

Unfortunately no way has yet been found of overcoming problems in authorizing users to execute commands to alter profiles for tapes. Only the owner or the creator of such profiles may execute the commands as long as the profile indicates that access to the data set is controlled by the default profile. The difficulty in overcoming the problem exists because no exit is entered when a command to alter a tape profile is executed. The problem could be overcome by coding the CLISTs as commands.

(b) Error messages

At this computing centre, the TSO command PROFILE WTPMSG has been included in a CLIST executed at every LOGON to cause operator messages to be displayed at time sharing terminals. Normally RACF messages would not be displayed since they are write-to-operator (WTO) messages.

8.6 The DD DATA JCL statement

The DD DATA statement creates a security exposure for the passwords of batch jobs read through a card reader (that is, jobs in the form of card decks). This computing centre uses the SMF job validation exit (IEFUJV) to prohibit the use of the DD DATA statement in such jobs, except under special circumstances. However, this has not presented a problem for users.

8.7 Creating data sets for other users

The existence of a default profile for each RACF user in this computing centre allows a slight relaxation of the rule that no user may create a data set for another user. The RACDEF exit allows such requests if a user has ALTER access authority in the default profile of the future owner of the data set to be created.

Even without this, the future owner may rename or copy a data set with appropriate authority or pre-allocate a data set to be loaded by the originator of the data.

The relaxation of the data set creation rules has removed the need for establishing large numbers of artificial RACF Groups in this computing centre, thereby reducing administrative and user education requirements.

8.8 Data set statistics

The data set statistics maintained by RACF have not been exploited in this computing centre because to find out when a particular user accessed another user's data set the second user would have to notice when the access count was incremented. The SMF record of access is more useful because the time and date and actual level of access are recorded, not just the maximum permitted level.

If part of the reason for producing reports on access to data sets is to monitor the reliable operation of RACF, then it is doubtful whether the RACF SMF records that identify accesses should be used. MVS also can produce SMF records describing data set accesses but these records are not complete for concatenated data sets.

In this computing centre, SMF records for concatenated partitioned data sets are produced by an SMF job validation exit (IEFUJV) which scans the JCL of batch jobs. This means that the records are always produced, whether the data sets are opened or not. Currently records are not produced for dynamically allocated concatenated data sets, although the dynamic allocation validation exit could be used for this purpose. As with the IEFUJV exit, the records would be produced whether the data sets were opened or not.

9. EXCEPTIONS IN THE USE OF RACF

There are several users at the Defence Research Centre that have special requirements not consistent with the security philosophy of RACF. Code has been included in various RACF exits to isolate these users from the remainder of the user population, and to restrict the functions they may perform, thereby maintaining the high level of security demanded by the Centre.

9.1 External users

Certain users should not be allowed even READ access to data sets owned by other users in spite of such access being granted, for example by setting the universal access (UACC).

This has been achieved by creating a RACF Group, XTN, to which these users are connected. The RACHECK exits have been modified so that when a user connected to this Group attempts access to a data set the access is never allowed unless it is his own or a system data set. The normal access available to system data sets is provided.

9.2 Special purpose terminals

A number of terminals are used for particular applications where each individual user is not identified to the system. For instance a terminal may remain permanently logged on although various people use it.

A special RACF Group, NOL, has been created to accommodate applications of this kind. The RACHECK exits prevent access to data sets other than their own and system data sets for users connected to the Group NOL. The RACINIT exits allow logon for the users without entering a password and prevent the execution of batch jobs. RACF terminal protection is defined so that any user may normally access any terminal but users connected to the Group NOL may only access a terminal if specifically permitted to do so by a RACF definition.

9.3 Mini-computer simulating an RJE terminal

A mini-computer is used for engineering design by several workshops and drawing offices. Many terminals are connected to the mini-computer and jobs are submitted to the central computer by the users to transfer data sets between the mini-computer and the central computer. Complete security or privacy is not provided in the mini-computer so that users could find out each others' passwords by inspecting the jobs which are built to be transmitted to the central computer.

Since data security in the mini-computer is incomplete, it is illogical to provide data security between users of the mini-computer for the data stored by them on the central computer. However normal security protection is required for their data relative to other users of the central computer.

The solution which has been evolved is to assign all such data sets to the special user WMD. WMD jobs will not require a password but it will only be possible to submit them from the identifiable mini-computer, not from any other terminal.

The implementation technique involves modifying JES2 to place the reader name in columns 73 to 80 of the job card image of a job (these columns previously contained the JES2 job number). The IEFUJV SMF exit (job validation) checks the terminal name for the userid WMD and cancels the job if it came from the wrong terminal. Appendix IX contains the details of the JES2 modification.

The user WMD is connected to the RACF Group NOL and thus does not require a password on jobs, may only access WMD data sets plus system data sets and is not permitted by RACF to logon at any terminal. The RACINIT exits have been extended to allow batch jobs from the user WMD even though connected to the Group NOL.

9.4 Service group processing data for many other users

One section of the Defence Research Centre processes data from trials conducted by many other sections. Various members of the above section need to create and modify data sets for these other users. A large number of processing programs, JCL, and CLISTs is maintained. Previously a number of userids were used to store the programs and submit processing jobs.

The solution has been to give each member of the section a personal userid and to change the userids used to prefix data sets containing programs, JCL or CLISTs into RACF Group identifiers. Personnel responsible for program maintenance are given appropriate access to the various Group data sets.

As well as this, all members of the section are connected to some of the RACF Groups. When data for another section is to be processed, that section will give the necessary level of access to the appropriate RACF Group, thus ensuring that any user connected to the Group will have the ability to process the data. Users connected to the Groups are given READ access to the data sets containing processing programs and procedures.

9.5 Special purpose data base enquiry terminal

A dedicated terminal is used to make enquiries into and also update a particular data base (using interactive programs under TSO). The terminal is sometimes unattended and is used by a large number of people who are not registered as users of the central computer.

The solution to this problem is to provide a userid, SUP, which is connected to the RACF group NOL and therefore is not allowed to log on at a terminal unless specifically permitted, may not submit batch jobs, and does not require a password to log on. The RACHECK exits have been extended to prevent SUP from gaining greater than READ access to any data sets including its own. Access is limited to SUP and system data sets plus the data sets of another RACF Group (ADP). RACF prevents attempts by SUP to log on to any terminal other than the single dedicated one. All updates to the data base are now done by users connected to the ADP Group with the appropriate level of RACF authority.

9.6 Typing pools

Several typing pools exist and their supervisors need to control text data sets which are being created.

A RACF group has been created for each typing pool and each typist has been registered as a computer user and connected to the appropriate Group. The supervisors have been given ALTER access to Group data sets but individual typists may only access data sets which they need to update. The supervisor will allocate any new data set and give the typist concerned UPDATE access to the data set.

9.7 Simulation task with several unidentified users logged on

A section of the Defence Research Centre runs a task which involves several users logged on to TSO who interact with each other and with a model via a number of data sets. The users of the model are not defined as users of the central computer.

The solution is to make either a RACF Group or one of the members of the modelling section the owner of the data sets. The person who supervises the use of the model will own an appropriate number of extra userids which he will use to log on for the users of the model. These extra userids will be given appropriate access to the data sets which they need to access - for example READ access to all the programs and UPDATE access to data sets which are modified. More than one person in the section will need a set of the extra userids because of possible illnesses or vacations.

10. STANDARDS THAT SIMPLIFIED THE RACF IMPLEMENTATION

Standards adopted by this computing centre when it first obtained an IBM 370 computer system helped in the implementation of RACF.

10.1 Userids

All userids are three characters long. This standard has been extended to RACF Group identifiers and has helped simplify the coding in the RACF exits.

10.2 Data set names

All non-VSAM data sets are prefixed by the userid or groupid of their owner. This is the naming convention assumed by RACF and therefore avoided the need for complex coding in the RACF exits to simulate it.

VSAM data sets are prefixed by a four character qualifier - the three character userid plus the character 'V' (indicating VSAM). However the RACF exits use only the first three characters of the dataset name to establish the userid of the owner, so these names still appear to conform to the naming conventions. This feature was extended as part of the RACF implementation to allow certain users and Groups to use qualifiers of three or more characters to prefix their dataset names, as long as the first three indicate the userid or groupid. For example, the RACF Group

IMS has data sets with several different prefixes, each representing a different component of the IBM IMS (Information Management System) product. Some of these are IMSVS (IMS system libraries), IMSLOG (log tapes), IMSDICT (IMS Data Dictionary), and so on. Datasets prefixed by any one of these qualifiers which are not specifically defined to RACF are all protected by the default profile that applies to the entire IMS Group.

This feature is particularly useful to RACF Groups, such as IMS, which have a large number of data sets that can be categorized into different areas of responsibility or function, for example. It enables the personnel responsible for these data sets to more easily recognize and therefore maintain them.

The names of the data and index components of a VSAM data set are also governed by a computing centre naming convention. The names must be the same as the cluster name of the associated data set, but with '.DATA' or '.INDEX' appended, respectively. This convention is used by RACF in two places. The first is in the CLIST that modifies the access available to data sets. Whenever a VSAM cluster name is processed the CLIST performs the same action on the data and index components, thereby avoiding the need for separate commands to be issued. Secondly, whenever one of the programs of the archiving scheme processes a VSAM data set through the RACHECK or RACDEF macro, they also perform the same action on its components, thereby ensuring that integrity is maintained.

10.3 Jobnames

The names of all batch jobs must be from four to eight characters long, and the first three characters must indicate the userid of the submitter. This information is used by the RACINIT exits to avoid the need for the USER parameter on the JCL JOB statement.

11. HISTORY OF THE DESIGN AND TESTING OF THE EXTENSIONS TO RACF

The concepts described in the implementation plan (Appendix I) were developed during August through October 1978.

The RACF exits were designed in November 1978 and three users were defined to RACF for tests. The exits were coded, tested and installed in December, 1978 and thirty users were defined to RACF to allow more extensive tests. However data set protection was not invoked. The design of CLISTs to replace the RACF commands was commenced (Appendix II).

The RACINIT return code and abend code had to be reset in the post-processing exit for batch jobs from users not defined to RACF for which the SUBMIT command generated the USER parameter on the JOB card. Otherwise RACF did not allow the job to execute.

During January and February 1979 disk data set protection was activated for three users and most of the problems in the exits were resolved. The CLISTs were coded and tested and all users were defined to RACF.

In March 1979 disk data set protection was activated for thirty users and tape protection for five users. At this stage care had to be taken that other users were not affected since they had not yet been informed that RACF was being installed - access to data sets had to be provided as required.

Protection for system data sets was activated in April 1979 - appropriate access had to be provided for users.

During May 1979 users were trained and were able to set up access authorities to their data sets in advance of activation of protection. All disk and tape data sets were protected in June 1979 and most problems of access had been resolved in advance.

A minor problem was caused by allowing the commands to be issued in advance. Because the RACF protection was not yet turned on in the DSCBs of data sets belonging to these users, the deletion or renaming of a data set did not cause the deletion or alteration of the RACF profile of a specifically defined data set. (Obviously no problem existed in the case of a data set not specifically defined). To overcome this, a program was written to check for occurrences of data set profiles in the RACF data set for which no data set existed on disk or in the archives. Exceptions, of which there were few, were repaired manually after consulting the users.

Few problems existed in the extensions to RACF because of the extensive testing which had been done. Also RACF has shown very few bugs and none of these has resulted in a security exposure.

Some peculiar effects were observed due to the way RACF maintains the duplicate data set backing up the primary RACF data set. A code can be set to ensure that all changes to the primary data set are copied to the backup. However, the physical organization of the data sets can change because of differences in timing of different changes while preserving the same logical content. Also the data sets are only enqueued SHARE while updating statistics so that statistics may not be maintained correctly.

12. HISTORY OF THE USE OF RACF

Presentations were made to all users in May 1979 to explain the use of RACF. Users were encouraged to set up access authority to their data sets in advance by using the commands provided. This was made possible by the way the commands were designed. A data set access report was presented to each user together with a description of how to use the commands. Each user's access report showed the data sets owned by other users which he had accessed during the previous six months, and the level of access to each. It was then the responsibility of each user to make sure that the owners of the data sets arranged appropriate access authority for him.

Protection was turned on for all disk and tape data sets plus those in the archives in June 1979. Users encountered few problems because most had already set up access authorities to their data sets. No cases have been reported where failure of protection occurred.

The impact on performance has not been measurable even though all data sets are protected. The inconvenience to most users has been minor because of the basic transparency of RACF for a user's own data sets. The uniform treatment of tape, disk and archive data sets and the use of the default profiles have also simplified the use of RACF.

The operational and administrative maintenance of RACF occupies trivial human resources.

13. CONCLUSIONS AND RECOMMENDATIONS

RACF would in its standard form satisfy most of the requirements of this computing centre for a software security package. RACF with the extensions and other security measures described in this report fulfills all the requirements. In addition, RACF has caused no system problems and no security exposures have occurred due to the failure of RACF.

We believe that IBM should address the problems in the use of RACF that are described in this report. Three possible improvements which are thought to be most important are summarized below. Disk tracks which are written on by a user and then freed for allocation to other users should be automatically made unreadable until written on again. The method of invoking RACF for controlling access to data sets stored on magnetic tape should be made as similar as possible to the method used for disk data sets. It should be possible, as a standard feature, to use a default RACF profile to control the access to a user's data sets and avoid the need to define a RACF profile for every data set.

GLOSSARY

access - used to indicate the use of a resource.

access authority - the type of access which a user may have to a resource.

archives - in this computing centre, disk data sets are regularly copied or archived to magnetic tape to provide free disk space. The archived data sets are managed by software which allows them to be retrieved to disk or deleted from the archives.

authorized program - a program authorized to perform any supervisor function.

audit trail - record of data set usage.

batch job - program executed by being scheduled from a queue of jobs which have been submitted at some previous time.

BPP - bypass password protection - allows a program to access protected data sets without authorization checking.

CLIST - TSO command procedure - a list of TSO commands which can be executed by entering a single command.

default profile - in this computing centre access to data sets is controlled by a default profile for each user unless the user defines a specific profile for the data set.

disk data set - a data file uniquely named (within this computer system) and stored on a direct access storage device (disk). All data sets stored on a disk are directly accessible by the computer system.

DSCB - record in the VTOC of a disk describing the location of a data set or of free space.

exit - a computing centre written routine called under defined conditions by a component of the operating system.

GDG - generation data group - automatic control and labelling of generations of data sets relative to the latest version.

GIS - query and report generation system.

Group - RACF facility to allow users to own common data sets.

Groupid - identifier of the Group data sets.

IMS - Information Management System - a data base management system.

INOUT - an OPEN parameter requesting that a data set be opened for input and output.

JCL - job control language - control statements used to describe the data sets, running options and programs required in a batch job.

JES2 - job entry system - controls the submission, scheduling of execution, and output of batch jobs.

JFCB - MVS control block describing the characteristics of an allocated data set and including the data set name.

macro - an assembler statement expanded by the assembler to include a number of machine instructions in a program.

magnetic tape - a data set may be stored on a magnetic tape which must be mounted on a tape drive by the operator to use the data.

MVS - the operating system used in the DRCS computing centre.

OPEN - the operation performed by the operating system before a data set can be used for input or output.

password - several alphanumeric characters known only to a user and the system which validates his identity.

profile - definition to RACF of the level of access available to a resource controlled by RACF.

RACF - Resource Access Control Facility - software package used to control access to data and to the computer system.

RACF Level - a RACF parameter available for use by a computing centre to further classify resources.

SMF - System Measurement Facility - records information about processes ocurring in the computer system.

STAIRS - library information retrieval system.

started task - program executed by an operator start command.

SUBMIT - TSO command used to cause batch jobs to be queued for execution.

SVC - an SVC machine instruction causes an interrupt which is handled by the operating system to give control to the supervisor routine requested in the SVC instruction. SVC routines are the part of the operating system used to perform functions for users.

SYSLOG - system log - a record of operator console messages and commands.

TSO - the time sharing system - supplies editing and program checkout facilities to interactive terminals.

universal access authority - the type of access to a resource which is permitted to all users.

userid - string of alphanumeric characters that uniquely identifies a user.

VSAM - virtual storage access method - the current IBM access method for indexed data sets.

VTOC - volume table of contents of a disk - each disk contains a VTOC which contains DSCB records describing the locations of data sets and free space on the disk.

REFERENCES

No.	Author	Title
1	IBM	"OS/VS2 MVS Resource Access Control Facility (RACF) General Information Manual". Version 1, Release 3, Form number GC28-0722-4
2	IBM	"OS/VS2 MVS Resource Access Control Facility (RACF) Command Language Reference". Version 1, Release 3, Form number SC28-0733-2
3	Gwatking, J.C.	"Automatic Migration of Data between Disk and Tape". 7th Australian Computer Conference, Perth, 1976
4	Gwatking, J.C.	"A Data Migration Scheme for the 370/168 Computer at WRE". WRE Technical Report 1870(A), June 1977
5	Gwatking, J.C.	"An Efficient Application of Disk Storage to the DRCS Data Migration Scheme". DRCS Technical Report ERL-0009-TR, April 1978
6	Goddard, P.N.L, Evans, H.H., Benyon, P.R., Lamb, D., Bennier, D.J., and Hore, G.F.R.S.	"Future Requirements for Computing Facilities at WRE". WRE Report WRE45, August 1970
7	Evans, H.H. and Goddard, P.N.L.	"The IBM 370/168 Computer System at WRE - The First Sixteen Months of Operation". WRE Report 1866(A), August 1977
8	Evans, H.H., Goddard, P.N.L., Helliwell, G. and Wilson, J.M.	"DRCS Computer System Survey of Requirements for the Period January 1978 to December 1980". DRCS Report ERL-0001-RE, April 1978
9	Gwatking, J.C.	"A Selective Data Set Backup Scheme for the DRCS Central Computer". DRCS Technical Report ERL-0021-TR, July 1978
10	Gwatking, J.C. and Collier, R.W.	"Management Procedures for Controlling Data Storage on the IBM System 370 Computer at DRCS. DRCS Technical Report ERL-0055-TR, January 1979
11	IBM	"OS/VS2 MVS Resource Access Control Facility Installation Reference Manual". Version 1, Release 3, Form number SC28-0734-2
12	Roughan, J.L.	"WRE Computing Centre User's Guide". WRE Manual 1492(AP), November 1976

APPENDIX I

PLAN FOR RACF IMPLEMENTATION AT DRCS

This Appendix contains a document prepared in September 1978 as a preliminary specification of the requirements and implementation of RACF at DRCS. Many of the ideas were later refined and modified during the detailed design and development phases of the project, as greater familiarity with RACF was obtained.

The Appendix is included in this report partly as a record of the complete documentation of the project and partly because it is interesting to compare the preliminary design with the final.

I.1 Principles in order of priority

(a) Ensure full IBM support and responsibility for security and integrity.

(b) Supply an effective level of security and integrity.

(c) Minimum impact should be caused to users.

(d) Implementation should be as simple as possible.

I.2 Specification of functional requirements

(a) Disk and tape data sets should appear to be treated identically by RACF (accepting that all data sets on a single tape volume will in effect have the same protection as that given to the last data set specifically protected on that volume). If the tape data sets are not specifically protected then they should have a default level of protection set by the user for all his data sets. Multivolume tape data sets should be protected as for single volume data sets. It should be possible to protect GDG data sets using just the GDG base name. (This is not feasible for tape GDGs).

(b) All data sets (tape and disk) should be automatically protected by RACF initially at a default level specified by the owner in his default data set protection profile. Any data set can optionally be given its own different protection attributes. The default profile should be easily altered by the user and the protection attributes of any data set which is not specifically protected should follow the change in the default.
The default profile for each user should initially be set up to allow no access to his datasets by all other users. Prior to actual protection of the data sets, each user should be given a report showing which data sets owned by other users he has been accessing. It will be up to him to make sure the owner authorizes future accesses to these data sets.
GDGs should get the default protection profile if the GDG base is not specifically protected.

(c) The archiving system should function without a significant increase in restrictions and with an archived data set having the same protection as it would have if it were still on disk. A retrieved data set should have the same protection as it previously had if specifically protected. Otherwise it should change its protection if the default profile has changed. ASCRATCH (deletes an archived data set) should only be possible with appropriate access authority for the data set.

(d) Job submission from TSO should remain simple except that specification of a user's Group will be necessary if the Group for the job is different from the Group to which he connected during LOGON.
Job submission on cards will require the addition of the PASSWORD and possibly GROUP to the job card unless the default Group is satisfactory.
LOGON will require specification of the GROUP if the user's default Group is not appropriate.
When all the passwords are in the RACF data set instead of the UADS data set, then we may allow user access to UADS only to add account numbers (to remove the need for them to be entered at LOGON).

(e) Operational maintenance programs should function normally (but it should be possible to subsequently reduce the authorization of each of these systems to the maximum which it requires). Inconsistencies in the RACF data set should not occur due to the activities of operational maintenance programs. In particular bypass password protection would cause the RACF data set not to be updated when programs running with this attribute cause additions, deletions and relocations of data sets.

(f) FORTRAN may have to be modified to only OPEN INOUT for a data set which is not write protected. OPEN INPUT would have to be used for a data set for which only read access is allowed. (FORTRAN now opens all data sets INOUT which would cause problems with read only data sets).

(g) The RACF command language reference manual contains descriptions of too many forbidden commands and operands to be suitable for even Group administrators, let alone ordinary users. An edited version of this manual should be produced at DRCS and additional features provided here should also be described in the new manual. The main addition should describe the use of default profiles to gain default protection for all data sets not individually protected.

(h) IMS data sets should initially be protected against access by other programs and when release 1.1.5 is installed the full protection features should be usable.

(i) Definition of project oriented groups of users should be done by CS Group. Each of these groups should be able to have a default profile to give data sets default protection attributes just as occurs for individual users. The Group administrator should have CONNECT authority for the Group and should be the only person able to change the access attributes for the default profile. The members of a Group should be given appropriate access authorities to Group data sets by the Group administrator.

(j) Sufficient backups of the RACF data set should exist so that complete recovery is possible under all eventualities. It is postulated that we will only run without RACF under very unusual circumstances.

(k) User reports should be generated to list accesses and attempted accesses to data sets.

I.3 Specification of implementation

(a) Users

Each user will be defined to RACF:

```
ADDUSER(userid) NAME(username) PASSWORD(current psswrd) GRPACC
ADSP DATA('address and tel.no.')
```

The userid and password will be obtained from UADS and the username, address and telephone number will be obtained from the data set containing names and addresses. The ADDUSER commands will be automatically generated by a CLIST.

PROFILE WTPMSG will be issued for each user in the system LOGON CLIST to cause RACF error messages to be issued to TSO terminals.

Each user will be given a default data set profile:

```
ADDSD 'userid.RACF.MODEL.PROFILE' UACC(NONE) NOSET AUDIT(FAILURES)
UNIT(DISK) VOLUME(DUMMY)
```

The user may change the profile e.g.

```
   ALTDSD 'userid.RACF.MODEL.PROFILE' UACC(ALTER)
or PERMIT 'userid.RACF.MODEL.PROFILE' ID(XYZ ABC) ACCESS(READ)
```

(b) Disk data sets

When an attempted access to a data set occurs the RACHECK preprocessing exit will bypass further checking if the userid is the same as the first level qualifier of the data set name. Otherwise if a disk data set is defined to RACF normal checking will be done. If the disk data set is not defined to RACF, then the RACHECK post-processing exit will substitute the name of the default profile for the data set to be checked and cause RACHECK to be reinvoked. Then the default profile will be used to provide the default access authority for the data set. If a profile for a GDG base exists then it will be used, (caused by the RACHECK preprocessing exit) otherwise the default profile will be used.

The RACF commands ADDSD, ALTDSD, DELDSD, LISTDSD may be used directly to create specific protection profiles for individual data sets, modify them, delete them, or list them. The command exit will have to be used to allow the NOSET operand of these commands to be used for group data sets or for other data sets to which ALTER access is available since all data sets will have the RACF DSCB indicator turned on.

The PERMIT command will not work for a data set which is not specifically defined to RACF unless a definition is created by an exit in this case. It is probably unnecessary to do this as a user can easily define the data set to RACF using ADDSD or a CLIST that we might provide to perform the same function which would merge the new attributes and the default attributes.

A CLIST could be created to combine the functions of all the RACF commands and deal with the problems when profiles do not exist for data sets.

The SEARCH command will only list those RACF protected disk data sets which have been specifically defined to RACF. This should be reasonable since only the more sophisticated users will use the SEARCH command.

NOTE

It has been decided not to use RACF statistics since the SMF type 14 and 15 records are currently produced for the backup system, tape management system, archiving system, and access list reports. It would involve a great deal of work to modify these systems and the equivalent of or better than the RACF statistics are currently produced. However the RACF audit records indicating changes to the RACF data set and unsuccessful access attempts will be produced.

Since statistics are not to be used it does not matter that RACHECK will be bypassed in some cases or that every data set does not have a RACF definition - either of these conditions prevents the recording of statistics.

Ultimately it will be desirable to use RACF audit records instead of SMF type 14 and 15 records since IBM is more likely to support the RACF records properly.

(c) Tape data sets

The introduction of protection for tape data sets may be delayed until a later stage.

Tape data sets which are not specifically protected will use the default profile for disk data sets.

When a data set on a standard label tape is created the RACHECK post processing exit will determine if a profile already exists for the volume or volumes. If not, the exit will create one for each volume by issuing a RACDEF macro and then place the userid and a one-byte flag in the installation data field. The UACC will allow any access. If a profile already exists for the volume and the userids match, the request will be allowed. If the userids do not match and the flag byte in the user field is set (which means the default profile should be used), then the check will be repeated against the default data set profile.

The checks performed for a read access are the same as those for a write access when the profile already exists.

Thus a tape data set will use the disk data set default access authority if no specific access authority has been defined for the tape.

The CLIST mentioned above in (b) will also execute RALTER and PERMIT commands for tape volumes where the user specifies the data set name. A catalog search will provide the CLIST with the volume serial number and the flag in the installation data will be set by the CLIST to indicate whether or not the default profile is to be used.

Specific protection of a GDG base where the data sets are on tape will not be possible. Either the default profile will have to be used or each generation will have to be specifically protected.

Note that since there is never a RACF definition for a tape data set but only for a tape volume, each data set on a tape (if there is more than one) will have the same access authority, namely that last defined. This is consistent with the fact that access to all of a tape is possible once access to one data set on the tape has been achieved.

When all the datasets on a tape have been deleted it will be erased and returned to the scratch pool for reuse, as now. The erase program will be authorized and will delete the profile associated with the tape volume.

(d) Archiving

All archive tape volumes will be RACF protected with universal access authority of NONE and owner OPS. When a data set with a specific definition in the RACF data set is archived, the archiving program will modify the volume serial number in the definition to ARCHIV. The reverse will happen on retrieval. If a data set is backed up, a duplicate definition will be created with ARCHIV as the volume serial. Reload will operate in a similar manner.

If a data set is scratched from the archives, then a specific definition for volume ARCHIV in the RACF data set will be deleted. The archiving software will be privileged and thus will bypass the protection of the RACF tapes and the normal checks performed for protected data sets. Each program will therefore have to perform its own authorization checking to ensure that the user is permitted to perform the requested function on the data sets. The user will need ALTER authority for any deletion, which includes ASCRATCH, ARENAME, as well as when another version of a data set must first be deleted in order to carry out a RETRIEVE, RELOAD, ARCHIVE or BACKUP request. These four commands will also require READ authority for the version of the data set they are to transfer between the archives and disk. The EXPIRY and MIGRATE commands will require no authorization.

(e) Batch job validation

The RACINIT exit will get the userid from the first 3 characters of the jobname so that the USER field on the job card will be unnecessary. The PASSWORD will have to be added to all job cards but TSO submit will add this to submitted jobs. RACF will use the default Group of a user if GROUP is not specified. TSO submit will add the logon GROUP to a job card. The logon GROUP will be the user's default Group if unspecified.

The command package will add PASSWORD, USER and GROUP to jobs with no job card. We may need to modify our SUBMIT exit to do this for jobs which have a job card included.

(f) TSO LOGON

The logon will be the same as now except for the addition of GROUP if other than the user's default Group is required, and the requirement to change the password at intervals. The maximum interval between password changes will be set at 90 days.

Since logon passwords will be in the RACF data set, the UADS data set will no longer be important for system security. Thus it may be possible to allow users access to the UADS data set to insert accounting information, thus avoiding the need to enter it at every logon. Simple CLISTs could be provided to add, change and delete accounting information. It would be a good idea to remove information on the ACCOUNT command from HELP so that users would not be able to find out how to modify other aspects of their user attributes.

(g) Operational maintenance programs

The started task which is used to submit maintenance programs to the internal reader will not need a password, and does not normally have access to any password. However the submitted jobs must have passwords on their job cards so that some way must be found to get the password for a userid out of the RACF data set. Of course this could only be done by a job with authorization to read the RACF data set.

An alternative might be to mark such submitted jobs in a way which would allow the RACINIT exit to recognize that there was no need for a password. Such a method could be a security loophole since any user who knew the technique could submit jobs without supplying the correct password and thus gain access to any part of the system without detection.

Another method which is both practical and secure would be to only allow logon or job start for users who have higher than the normal authorization if confirmed by the operator. Thus a password for such jobs would not be required.

Another solution would be to store the OPS password in a protected data set and automatically and randomly change it every day at IPL. OPS tasks would be able to read it from the data set to submit other jobs.

Assuming that the above problem can be resolved, either by implementing one of the suggested solutions or inventing a better one, it is proposed that initially the userids of the submitted maintenance programs be given the highest authorization possible to ensure that they work. Later the authorization will be reduced to the maximum required. If bypass password protection is required the program concerned will have to update the RACF data set appropriately since this will also be bypassed.

Some maintenance programs, running as batch jobs, also generate and submit other jobs to the internal reader. Batch jobs therefore also need a means to determine their own password dynamically so they can insert it on the generated job cards. One solution would be to provide a routine which a program could call and which would return the password and userid of the caller. During RACINIT processing the password could be stored in the user's address space for later reference by the routine. There is no reason why such a routine could not be made generally available to all users.

It is proposed that password protection and not RACF protection be retained for SYS1.OPSAUTH (the library containing authorized and privileged utilities) since the operator should continue to be involved whenever this data set is accessed.

In the future, this case, and the expiry date protection mechanism which requires operator authorization for modifications, could be simulated by additions to a RACHECK exit. Any attempted modification to a SYS1 data set or read access to OPSAUTH could require an operator reply to authorize the access. The user would also need to be authorized within RACF to access such a data set. It is not intended to implement this proposal initially.

The cleanup program should list the names of any data sets which are not RACF protected (the DSCB indicator is off).

It is possible for any user to prevent access by specific other users e.g. operations. This would be a nuisance but the most sensible way to overcome it should be by administrative methods if it ever occurs.

A CSECT has to be built with the names of all the started procedures.

(h) Operational precautions

The use of BLP (bypass label processing) for tape will have to be carefully controlled, as it is now.

The use of DD DATA in a job read from a card reader presents an exposure as a user might gain access to all jobs following his on the reader if he omits the end of file delimiter. The IEFUJV exit will have to be modified to convert DD DATA to DD *. This will prevent any subsequent jobs from being destroyed as well as prevent a privacy exposure.

A data security exposure exists now because anybody can delete a data set catalog entry even if the data set is password protected. With RACF it is possible to protect the catalogs (with UACC of UPDATE) and RACF prevents users from manipulating, changing, or creating catalog entries for which they do not have ALTER authority. This is not documented in any RACF manual.

(i) FORTRAN

Most users will probably require default protection of READ but no WRITE. This allows other users to read their data sets. FORTRAN always opens a data set for INOUT, even when only input is to be performed. This would cause an access failure to a WRITE protected data set. The users can solve the problem by specifying input only on DD statements or in ALLOC-ATTR but this is rather cumbersome. It is proposed that the FORTRAN OPEN routine be modified to only open INOUT when there is no write protection. Otherwise it would open for input only. The RACHECK macro would be used to check the access authority. IBM are investigating whether this has been done elsewhere. For tape data sets the check will have to be performed against the tape volume on which the dataset resides.

(j) IMS

All data bases will be RACF protected against use by other than their owners and the normal IMS programs which support the use of the data bases. Full security will be attained with the installation of IMS release 1.1.5.

(k) Existing data sets

Existing data sets, tape, disk and archived, will initially be given the default access authority of their owners' default profiles which allow no access by any users. Users will be able to modify the access available to their data sets before the date on which they will become protected.

(l) RACF Groups

The exits will treat Group disk or tape data sets just as they do individual data sets i.e. each Group will have a default profile data set and a Group data set will acquire the attributes of the default data set if not defined explicitly to RACF. Normally only the Group administrator will be able to change the characteristics of the default data set. (Note that it is not possible to logon with a Group name as a userid).

User Groups will be added using the command:

```
ADDGROUP (group name) SUPGROUP(CSGROUP) OWNER(OPS)
```

A Group administrator will be appointed by the commands:

```
ALTUSER userid GROUP(group name) AUTHORITY(CONNECT)
ALTUSER userid DFLTGRP(group name)
```

Group administrators will add and delete members of groups:

```
CONNECT userid GROUP(groupname) AUTHORITY((CREATE)) GRPACC ADSP
                                         ((USE    ))
```

```
REMOVE userid GROUP(groupname) OWNER(userid)
```

(m) RACF data set recovery

It is possible to maintain a duplicate RACF data set so that a hardware failure allows processing to continue without interruption. However, a logical failure would presumably affect both data sets similarly and an alternate form of recovery would be necessary. It is proposed that the RACF data sets be backed up every night using the normal backup system. It seems that activity on the secondary RACF data set should be low since only changes need to be recorded and most data sets will not have an entry in the RACF data set. The primary RACF data set will be much more active since a search for an entry will be necessary for each data set accessed which does not belong to the user performing the access.

A sample RACHECK exit to allow access to protected data sets with RACF inactive has been obtained. This will be installed so that it can be optionally included with MLPA in an IPL to allow recovery procedures on RACF data sets with RACF inactive.

(n) User data set access reports

A report of accesses to data sets will continue to be generated from SMF record types 14, 15, 17, 18 and so on. The RACF audit records describing unsuccessful accesses will be added to the access reports.

(o) RACF options

The RACF system wide options will be specified by the SETROPTS command:

```
SETROPTS CLASSACT(TAPEVOL) TAPE DASD NOTERMINAL INTERVAL (90)
    NOSTATISTICS(*) NOINITSTATS AUDIT(*) SAUDIT CMDVIOL LIST
```

giving tape and disk volume protection, no terminal checking, a maximum of 90 days between user password changes, no RACF statistics, AUDIT SMF records of all changes to the RACF data set, and a list of command failures due to inadequate authority.

(p) Creating data sets on behalf of other users

The procedure will be to create a user or Group data set in the creator's userid or Group and then authorize the new owner of the data set to access the data set, e.g. to copy it he will need READ authority or to rename it he would need ALTER authority.

In reloading an unloaded data set from a distribution tape, it will be necessary in some cases to use the RENAME parameter of IEHMOVE to change the data set name to one's own dataset.

(q) Error message

The IEFU83 exit can supplement the 913 abend code with a TPUT message. This may be more acceptable than changing all the user profiles to get WTP messages. A sample exit has been obtained.

I.4 RACF installation program

October	install RACF design the implementation write exits, programs and CLISTs define the education required write the documentation define the operational policy
November	test the implementation educate the operators who will administer RACF define all users as inactive RACF users
December	test the implementation on CS Group educate duty programmers and the groups to be involved in the January tests
January	test the implementation on two other DRCS groups educate all users
February	introduce RACF for all users
March	introduce tape data set protection if delayed

APPENDIX II

COMPUTER BULLETIN NO. 122
NEW SECURITY AND PRIVACY FACILITIES (RACF)

This Appendix contains the DRCS Computer Bulletin sent to users to introduce RACF and related security measures. Included are descriptions of the TSO CLISTs SHARE (to define access to a data set) and LISTP (to list access to a data set).

II.1 Introduction and background

A new facility has been added to the IBM 370 computer operating system software which provides a much more powerful means of controlling access to data stored on the computer. It is known as RACF (Resource Access Control Facility) and is a fully supported IBM product. As more users and particularly as terminals from other laboratories and establishments are connected to the 370 system it becomes increasingly important to employ rigorous but flexible security techniques.

The new facility is very different from the existing arrangements and every effort has been made by Computing Services Group to minimise the number of commands that need to be understood and used. In fact, if you only wish to access your own datasets no change is involved. It is however important that you read at least the first 3 sections of this bulletin.

Until now all data sets were accessible to every user unless they had been individually password protected. Under the RACF system access to every data set is confined to its owner unless arrangements are made otherwise. The existing facility of password protection for individual data sets will be removed, since RACF provides equivalent function.

Since many users share data sets, it will be necessary to establish sharing arrangements before RACF is brought into effect. TSO commands have been provided to make this simple and users who access data sets belonging to others will be provided with a list of the data sets they have accessed during the last six months.

The system has been designed so that access to disk, tape and archived data sets will be controlled in the same way. Only the standard range of labelled magnetic tapes which are stored in the computer centre will be protected.

The security of all data sets under RACF depends on each user being positively identified when he logs on to the system. Therefore, logon passwords will be classified SECRET. The practice of sharing userids and passwords will not be allowed. If you have any suspicion that your password is known to others it must be changed immediately. It is now possible for you to change your own logon password easily at any time and in any case, to ensure its secrecy, you will have to change it every 3 months. To maintain a satisfactory level of security, a terminal at which you are logged on must not be left unattended.

In addition to the protection of data sets by RACF, a facility to print security classifications on job output has been provided. This facility is described in Section 6. The distribution of classified output is discussed in Section 7.

II.2 Implementation of RACF

The implementation has been planned to provide total protection for all data sets while causing the minimum of disruption. Protection for all your data sets will commence on 11/6/79, and this level of protection will prevent any shared access (either read or write) to your data sets unless you have previously taken action. The action must take the form of issuing commands to RACF declaring which data sets are to be shared with which users. The commands to set up access authorities to your data sets can be issued from 1/5/79, so that when protection is introduced no disruption will be caused to other users who need to access your data sets.

Your data sets can be shared in two ways. First, all your data sets can be shared with specified users (see example (c) below). Second, an individual data set can be shared with as many users as you like (see examples (a) and (b) below). If an individual data set is not specifically defined to be shared in this way then it is shared according to a default (for example as defined in example (c)). A default list of users to share data sets should be adequate for the majority of data sets owned by most users. We recommend that you attempt to create a default list of users to share all your data sets since this is simple and easy to maintain. The ways in which your data can be accessed can be displayed by a command (see examples (e) and (f) below).

Some examples of commands to give various levels of access are described below and a more comprehensive description is given in Section 5.

(a) to allow all users READ access to one of your data sets (READ access allows a data set to be input, copied or listed but not updated or deleted):

```
SHARE dsn UACC(READ)
```

(the data set name must include the type - for example .CNTL)

(b) to allow several users update access to one of your data sets (UPDATE access allows a data set to be written or updated but not deleted. UPDATE includes READ access - READ access is defined in (a) above):

```
SHARE dsn ID('userid1 userid2 ....') ACCESS(UPDATE)
```

(the data set name must include the type - for example .FORT)

(c) to allow several specific users a default access authority of ALTER to all of your data sets except those which are defined specifically by the SHARE command as in (a), (b) and (d). (ALTER access allows a data set to be read, updated and deleted. ALTER access includes UPDATE access and READ access):

```
SHARE * ID('userid1 userid2 ....') ACCESS(ALTER)
```

(d) to allow several users READ access to one of your datasets:

```
SHARE dsn ID('userid1 userid2 ....') ACCESS(READ)
```

(e) to display the default access available to all datasets not defined specifically as in (d):

```
LISTP *
```

(f) to display the access available to a specific data set:

```
LISTP dsn
```

To ensure that appropriate access to data sets is available, a list of the data sets owned by other users which you have accessed during the past 6 months is attached. It will be necessary for you to approach these users so that they may arrange access to their data sets.

II.3 Consequences of the installation of RACF

The rigorous application by RACF of the principle of only sharing data with authorized users will conflict with procedures that were previously legitimate. Also some features of the implementation of RACF need explanation even though great efforts have been made to design it in a consistent manner. Some consequences of the implementation of RACF are described in the following paragraphs.

(i) Archiving

RACF will prevent you from retrieving another user's data set from the archives unless you have READ authority to that data set. Other commands of the archiving system require ALTER authority.

(ii) Creating data sets for other users

To create a data set for another user, the data set is given a prefix equal to that other user's userid. For tape data sets, this is readily done, but should be followed by a SHARE dsn OWNER(userid) command to make the other user the owner of the data set. For disk data sets, you will need to be on the other user's default access list with ALTER authority. Alternatively, the other user can make a copy of your data set (for which he will need READ authority).

CLISTs should be checked to ensure that they do not use &SYSPREF as the prefix of any data set which they create. JCL should also be examined to ensure that data sets for other users are not created.

RACF does allow for the definition of Group data sets. This may be of interest to some groups of users - for example those associated with a project or task. All users connected to a RACF Group are allowed to create Group data sets and access the data sets. The groupid is the prefix of Group data sets but it is not a userid so it is not possible to logon with the groupid.

(iii) FORTRAN I/O

FORTRAN programs open all data sets FOR INPUT and OUTPUT so that a FORTRAN program which merely READs a data set normally requires UPDATE access authority to that data set. If the data set is yours, there is no problem, but if you have only READ access to another user's data set you will have to use the IN parameter of the FILE command or the IN subparameter of the LABEL parameter on a JCL DD statement. The IN parameter causes the data set to be opened for INPUT only so if a WRITE is attempted it will fail with an I/O error.

(iv) Batch jobs

All batch jobs will require your logon password on the JOB card but the SUBMIT command will add this automatically to jobs submitted from TSO. If the jobname contains another userid, SUBMIT will change it to your userid instead of rejecting the job as it does currently.

Card jobs will require the logon password on the JOB card in the format:

```
.........,PASSWORD=password
```

The password must be coded on a continuation card of the JOB card with printing suppressed. All card decks should be treated as if classified SECRET, since the security of all data sets will depend on the security of the logon password. To ensure the privacy of the password and to avoid accidental disclosure, the card containing the password will be destroyed by the operator as soon as a job has been read in at the central computer. A new card will have to be punched and inserted every time the job is submitted. The password will be printed as XXXXXXX on the job printout so that the listing need not be protected.

(v) Password changes

Your password will have to be changed regularly, but this is very easy to do. If you wish to change your password at any time it may be changed at LOGON to TSO or in a batch job (see below). At LOGON, enter:

```
oldpassword/newpassword
```

when prompted for the password.

If you have not changed it often enough, TSO will prompt you to enter the new password. The sequence of prompting is given here:

```
logon userid acct(nnnnnn/nnn)
ENTER CURRENT PASSWORD FOR USERID
old password
CURRENT PASSWORD HAS EXPIRED AND NO NEW PASSWORD ENTERED
REENTER
new password
```

If your first activity on the day the password needs changing is to submit a batch job on cards then the job will be rejected because the password needs to be changed. The job can be resubmitted with the old and new passwords in the format:

```
.....,PASSWORD=(oldpassword,newpassword)
```

If a job is not run on the day it is submitted (for example there is too much work) and the password is due to be changed on the next day then the job will fail because the password is no longer current. The job will need to be resubmitted.

(vi) GDG data sets

Disk generation data group (GDG) data sets may not be given different levels of access for different generations. All generations will have the same default level of protection as all other disk data sets which are not defined individually to RACF. On the other hand the GDG collection of data sets may be protected differently from the default by protecting the GDG base name. Note that if the GDG base is deleted the definition to RACF will not be automatically deleted and must be deleted using the command:

SHARE gdgbase DEFAULT GDG

GDG data sets stored on tape must either be defined to RACF for each generation using the full data set name (name.GnnnnVnn) or will be protected according to the user's default for all data sets not defined specifically to RACF.

(vii) DD DATA statement

The JCL statement DD DATA causes a security exposure, and therefore its use will, with the introduction of RACF, be prohibited. The DD DATA statement was used to process JCL statements as an instream data set. Therefore if you wish to enter JCL into a data set it will now have to be entered at a terminal by you or by the punch room staff.

(viii) Magnetic tape data sets

RACF protection of tape data sets is by tape volume so that different levels of access cannot be defined for multiple data sets on a single volume. All data sets on a volume are protected identically so that a definition to RACF of an access authority to any data set on a volume applies to all the data sets on the volume. Only the standard range of labelled tapes which are stored in the computer centre will be protected.

(ix) Partitioned data sets

The members of a partitioned data set cannot be given different access authorites since only the partitioned data set can be defined to RACF - not the members.

(x) Creation of sensitive data

Since a data set, when first created, is protected by the default access list defined by you, it may be necessary (for sensitive data) to preallocate a data set and specifically define no access to it before loading data into the data set.

(vi) Data set access reports

Every fortnight, a report is distributed to you showing which users accessed your data sets. The report shows the level of access, for example READ or UPDATE, and the number of times it occurred. After RACF becomes active, you should regularly check this report to make sure that accesses are consistent with your definition to RACF of how your datasets are to be shared with other users.

The content of the data set access report will be enhanced with a list of users who tried to access your data sets and failed because of RACF protection. In cases where this is not simply because of your omission to provide appropriate access to your data sets, you may wish to investigate why such an attempt was made. You can find out another user's name and address with the TSO command:

USER userid

II.4 Submitting batch jobs to the internal reader from a batch job

A small number of users have programs which submit jobs to the internal reader. The following subroutine and utility program assist in creating a job to be submitted to the internal reader by supplying the user's own password (needed for the JOB card of the submitted job).

(i) Subroutine PASSWRD

This subroutine may be called from a PL/I program to return a user's own password.

Calling sequence

```
DCL     PASSWRD ENTRY OPTIONS (ASM, INTER);
DCL     USERID CHAR(3),
        PASSWORD CHAR(8),
        LNGTH BINARY FIXED(31);
CALL    PASSWRD (USERID, PASSWORD, LNGTH);
```

The user's userid, password and the number of characters in the password are obtained.

(ii) Program OPSEDIT

This program is a replacement for IEBEDIT for submitting jobs through the internal reader. It finds any JOB cards in the input stream and adds the user's PASSWORD to them.

The JCL required is exactly the same as that required for the IBM utility IEBEDIT (see the OS/VS Utilities Manual, GC35-0005).

II.5 TSO commands for RACF

A user will control the access to his data sets by a default access list or by specifically defining to RACF which users may access an individual data set. Access to each data set on disk or tape will be controlled by the default access list when the data set is created. The user may modify the default access list or define the level of access to a specific data set by a TSO command.

The level of access available to any data set which may be defined specifically to RACF (differently from the default), consists of a universal access authority (UACC) and a list of specific users who are permitted access different from the UACC. The levels of access which can be defined are:

NONE - the user may not access the data set either to read, update or delete.

READ - the user may read or inspect the data set but not update or delete it.

UPDATE - the user may read or update the data set but not delete it.

CONTROL - equivalent to the VSAM control password.

ALTER - the user may gain any access to the data set (read, update or delete).

A default list of users and corresponding access authorites may be defined. Any user not on this list will have a default access authority of NONE to any data sets not defined specifically to RACF. This is equivalent to saying that the default universal access authority (UACC) is NONE.

When any data set is deleted, a specific definition to RACF of the level of access to the data set is also deleted. The definition will not automatically carry over to a data set of the same name that might subsequently be created.

A user not wishing to use TSO at a terminal may execute TSO commands in a batch job to authorize sharing of his data sets. See Computer Bulletin No. 100 for a description of how to execute TSO commands in a batch job.

SHARE command

The SHARE command is used to alter the access authority of all users or specific users to datasets or to provide a default access authority for datasets not defined specifically using the SHARE command. Most of the parameters of the SHARE command can be abbreviated.

```
SHARE {dsn | * }    [DEFAULT]    [UACC(uacc)]
SH

  [ID(userid) {ACCESS(access)|DELETE}]    [GDG]

  [FROM(dsn2)|FROMDEFAULT] [OWNER(userid)]    [ARCHIVE]
  [REPEAT]
```

dsn - data set for which protection is to be altered. The data set name must include the type qualifier - for example .FORT etc. (for VSAM data sets, the cluster, index and data components are dealt with automatically and identically - the dsn must be the cluster name).

* - alter default protection for all your data sets for which SHARE is not used to protect specifically. The parameter UACC is not permitted in conjunction with this parameter.

DEFAULT - remove specific protection from the data set - it will be protected according to your default.

UACC(uacc) - access authority to the data set for all users not specifically identified using the ID parameter. See the list of possible access authorities defined below. The UACC parameter is not allowed with dsn=* (the default).

ID(userid) - a user to be given a different access authority from the universal access authority (UACC). (A list of userids may be entered in quotes). The ACCESS or DELETE parameter must be used with the ID parameter.

ACCESS(access) - access authority for the user defined in the ID parameter. See the list of possible access authorities defined below. (If the ID parameter is omitted then the ACCESS parameter is changed to UACC by the SHARE command).

DELETE - the user defined by the ID parameter is to be removed from the list of users with specifically defined access authorities.

GDG - the dsn is a disk generation data computing centre base name.

FROM(dsn2) - copy the access list of users and authorities defined specifically for dsn2 into the access list for the data set. Note that the UACC defined for dsn2 is not copied so that the UACC for the dataset will be NONE unless it is explicitly specified.

FROMDEFAULT - copy your default access list of users and authorities into the access list for the data set. Note that the UACC defaults to NONE unless explicitly specified and also note that if you are protecting another users data set, it is his default access list which is copied, not yours.

OWNER(ownerid) - change the owner of the data set (only relevant for a Group data set). The owner of a data set is normally the creator.

ARCHIVE - the data set is in the archives (only necessary if another data set with the same name exists either on disk or tape).

REPEAT - if this parameter is specified the command will prompt for further data set names and add identical protection for each after they are entered.

Access authorities:-

NONE - no access allowed

READ - only read access

UPDATE - the data set may be updated but not deleted and the SHARE command may not be used.

CONTROL - the same as UPDATE for non-VSAM data sets - equivalent to VSAM CONTROL password for VSAM data sets.

ALTER - all forms of access permitted, including the use of the SHARE command.

LISTP command

The LISTP command is used to display the access authority of other users to datasets. Most of the parameters of the LISTP command may be abbreviated.

```
LISTP {dsn | * | (DISK) | (ALL)} [ID(prefix)]
LP

      [PREFIX(prefix)] [ARCHIVE] [GDG]
```

dsn - defines the data set whose protection attributes are to be listed. The data set name must include the type qualifier - for example .FORT.

* - the default protection attributes to be used for all data sets not specifically defined using the SHARE command are listed.

(DISK) - the protection attributes of all specifically protected disk data sets are to be listed. Tape data sets and data sets with the default protection are omitted. The command executes much faster with this option than with (ALL) - see below.

(ALL) - the protection attributes of all specifically protected data sets are to be listed. Data sets with the default protection are omitted. The LISTP command is very slow for this option.

ID(prefix) -
PREFIX(prefix) the protection attributes of specifically protected data sets to which you have access and which begin with the indicated prefix are listed. The prefix may include the userid plus one or more qualifiers of the data set names to be selected.

ARCHIVE - indicates that the data set specified is in the archives. This is unnecessary unless a data set of the same name also exists on disk or tape.

GDG - the dsn is a disk generation data group base name.

An example of the output of the LISTP command follows:

```
listp name.text
 INFORMATION FOR DATASET XYZ.NAME.TEXT

 LEVEL  OWNER   AUDITING  UNIVERSAL ACCESS
 -----  -----   --------  ----------------

  00     XYZ    FAILURES        NONE

 YOUR ACCESS  CREATION GROUP  DATASET TYPE
 -----------  --------------  ------------

 NONE GIVEN       DRCS        NON-VSAM

 VOLUMES ON WHICH DATASET RESIDES  UNIT
 --------------------------------  ----

 STOREA                            DISK

   USER    ACCESS   ACCESS COUNT
 --------  -------  ------------

 ABC         ALTER     00000
 QRS         READ      00000
```

Universal Access is equivalent to UACC in the SHARE command and indicates the access authority which all users have except those in the access list. The access list appears last and contains specific userids and access authorities. This list corresponds to the ID and ACCESS parameters of the SHARE command.

LISTUSER Command

```
LISTUSER
```

The details of your RACF user profile are listed.

PASSWORD Command

```
PASSWORD [INTERVAL(change interval)]
```

The command can be used to alter the maximum interval allowed between password changes. The interval between password changes may not be increased to a period greater than the computing centre standard which is currently 90 days.

II.6 Printing the security classification on listings

A facility now exists on the IBM 370 computer system for automatically printing the security level of classified computer printouts at the top and bottom of each page.

The security level can be selected individually for each output dataset produced by a job, and is indicated by the choice of output class for the printout. No other action is necessary. The three new output classes available are R for Restricted output, C for Confidential and S for Secret. All other classes are assumed to be unclassified, unless the user produces his own security messages.

In most respects classes C, R and S are treated the same as class A output. However, several lines per page are required for the security messages when using these three classes, leaving users with a maximum of 60 lines per page for their own output. Other output classes allow up to 66 lines per page (see TM 1662(AP)).

Users should be aware that the security classification messages are not incorporated into the output until it is selected for printing on a local or remote printer. Therefore, if the TSO OUTPUT command is used to scan the output at a TSO terminal prior to printing, the messages will not be present.

Several examples of using the new output classes follow.

(a) Userid ABC requires a batch job to compile and execute a FORTRAN program and produce printed results on logical unit 6. These results are restricted, but all other output produced by the job is unclassified. The job will be submitted from TSO and the results are to be held for scanning on TSO prior to printing. The JCL could be -

```
//ABCJOB  JOB  ,,CLASS=X,MSGCLASS=A
//  EXEC  FTG1CG
//FORT.SYSIN DD *
     FORTRAN program
//GO.FT06F001  DD SYSOUT=R,HOLD=YES
```

(b) A user runs a FORTRAN program interactively from TSO, and the job produces printed output that is confidential and is to be sent to remote printer RMT14.
The TSO commands to allocate FORTRAN logical unit 6 could be -

```
     ALLOCATE FILE(FT06F001) SYSOUT(C) DEST(RMT14)
or
     FILE FI(6) PRINT(C) DEST(RMT14)
```

(c) Userid ABC has a dataset named ABC.SECRET.DATA that contains data classified as Secret. He wishes to use the TSO PRINTOFF command to obtain a listing of the dataset at the central printer. The command could be -

```
     PRINTOFF SECRET.DATA CLASS(S)
```

II.7 Distribution of classified output

Distribution of classified output from the Computing Office will be controlled.

A log of classified jobs will be kept in the Computing Office and anyone collecting the output will have to sign for it. If someone other than the owner wishes to collect the output, they will need written authorisation which they can present to the Computing Office, e.g.

"I authorise A. Brown to collect 6 jobs CXDA - CXDF submitted
at 11 a.m. on 27/4/79.
C. Dale"

The listed job names plus date and time must give sufficient information to allow Computing Office staff to identify the output. The authorisation must be signed either by the owner or by the head of the section. The collector will be asked to sign for the output and should display his DRCS pass as identification.

Classified output directed to a remote terminal is the responsibility of the user creating it.

Unclassified output is not affected by the new arrangements.

APPENDIX III

INSTRUCTIONS ON THE MANAGEMENT OF RACF GROUPS

This Appendix contains a document distributed to administrators of RACF Groups at DRCS.

III.1 Defining the group

When a RACF group is established one user must accept responsibility for its administration. This user must approach L. Binns or G. Owen of the Operations Section of CS Group to define the necessary RACF environment. The definition includes the following functions :-

(a) creation of the group, with a mutually agreed three character name,

(b) creation of an initial RACF default profile for the group's datasets that are not specifically protected. This profile will include UACC(NONE), which cannot be altered, and will nominate the administrator as its owner,

(c) connection of the administrator to the group with CONNECT authority, which allows him to connect other users to the group.

III.2 Connecting users to the group

A user does not have to be a member of a group in order to access or create datasets belonging to that group (i.e. datasets having the group name as their first level qualifier). These functions are controlled solely by the access authorities granted in the group's default profile and those of any specifically protected datasets. The only advantage in being connected to a group is that it may be necessary in order to access certain datasets. This is because the access lists in dataset and default profiles may include group names as well as userids. Either may be specified in the ID parameter of the SHARE command. If a group name is included then any user executing under control of that group is granted access to the dataset, without the need for his userid also being in the list.

Before a user can gain access to a group he must first be connected to it by the administrator. The format of the command to do this is -

```
CONNECT userid GROUP(group-name) AUTHORITY(group-authority)
```

The group authority defines what functions the user may perform in the group and must be USE or CONNECT:

(a) USE

A user with this authority can access group datasets. The level of access available is that granted to the user in the RACF profile of a specifically protected dataset or in the group's default profile for one not so protected. The level may be NONE, READ, UPDATE or ALTER, which also allows creation when specified in the default profile. As already mentioned, these functions are also available to users who are not members of the group. The extra privilege granted to group members is that they can access datasets to which the group itself is authorized, under the circumstances described in Section 4.

(b) CONNECT

This authority is the highest available and is normally assigned only to the group administrator. It includes the functions of USE and in addition allows the holder to connect other users to the group and remove them from it. CONNECT authority could be assigned temporarily to another group member while the administrator is on leave, for instance, and revoked on his return.

For example, to connect user ABC to group XYZ with USE authority the command would be -

```
CONNECT ABC GROUP(XYZ) AUTHORITY(USE)
```

III.3 Altering the group activity

The administrator may alter the group authority (USE or CONNECT) of a user already connected to a group by simply re-issuing the CONNECT command.

III.4 Gaining access to the group

Under RACF each user must be connected to one or more groups, one of which must be designated his default group. In our group all users are in fact connected to the group DRCS, which is also the default, when they are initially defined to RACF. Being connected to a group does not automatically grant the user authority to datasets that mention the group name in their access lists. The user must also be executing under control of that group. All TSO sessions and batch jobs initiated by a user execute under his default group unless another group to which he is connected is specified in the GROUP parameter of the TSO LOGON command or the GROUP parameter of the JCL JOB statement. For example, for userid ABC to logon to group XYZ (not his default), the command would be -

```
LOGON ABC GROUP(XYZ) ACCT(123456/789)
```

This technique is obviously inconvenient for a user who normally wishes to access a group other than DRCS (the standard default group). Accordingly a TSO command is provided for any user to change his own default group, provided he has already been connected to the group. The format of the command is

```
DEFGROUP group-name
```

For example, to change the default group to XYZ the command would be -

```
DEFGROUP XYZ
```

The output from the LISTUSER COMMAND (see Computer Bulletin 122) indicates a user's current default group.

III.5 The group's default profile

When a group is first established the administrator is nominated as the owner of the default profile. He must assign ALTER access authority to all users who are permitted to create group datasets. The administrator and any other user with ALTER authority is then permitted to change the default profile as required. The sequence of commands necessary to achieve a change to the group's default is -

```
PROFILE PREFIX(group-name)
SHARE * other parameters
PROFILE PREFIX(userid)
```

The access list for the default may include group names (including the default's group), userids connected to the group and even userids not in the group. For example, suppose user ABC is the administrator of group XYZ and that all members of the group require ALTER authority in the default profile. In addition user LMN, not a group member, requires READ authority. The commands to achieve this could be -

```
PROFILE PREFIX(XYZ)
SHARE * ID(XYZ) ACCESS(ALTER)
SHARE * ID(LMN) ACCESS(READ)
PROFILE PREFIX(ABC)
```

This example illustrates that the group name, or alternatively the individual userids of the group members, must be mentioned in the group's default profile and the profiles of specifically protected datasets (see below). Access authorities to group datasets must be implicitly stated, even for group members. There is no feature similar to the explicit ALTER authority granted to each user over his own datasets.

III.6 Specifically protected group datasets

When all group datasets are initially created they are protected by the group's default profile. Any user with ALTER access authority in the default may specifically protect a group dataset, and that user becomes its owner. The specific protection may be changed or even deleted by the dataset owner or by any other user who currently has ALTER access authority to the dataset.

III.7 Listing users connected to the group

The group administrator may obtain a list of the userids connected to the group using the command -

```
LISTGRP group-name
```

III.8 Removing users from the group

The group administrator may also remove, or disconnect, users from a group when they no longer have a requirement to be associated with it. The format of the command is -

```
REMOVE userid1 GROUP(group-name) OWNER(userid2)
```

The OWNER parameter identifies another member of the group (userid2) who is to be assigned ownership of all specifically protected group datasets still owned by the user being removed (userid1). This parameter is not required if no such group datasets exist.

Note that the owner of each specifically protected group dataset is indicated in the output of the LISTP command and can also be changed by the current owner using the SHARE command.

If a user is disconnected from a group it may also be appropriate to remove his userid from the access list of the group's default profile and those of any specifically protected group datasets.

If the group administrator is being disconnected he must first nominate another member as the new administrator, by giving him CONNECT group authority. In addition he must assign ownership of the group's default profile to the new administrator using the SHARE command.

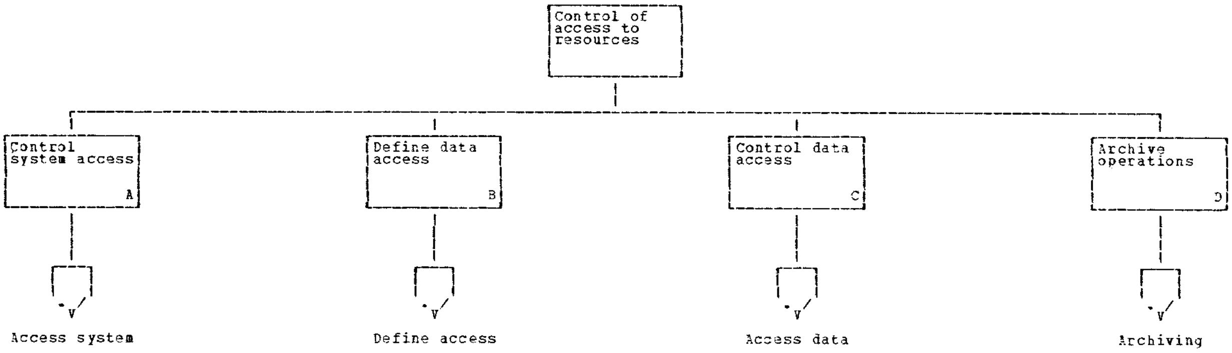
Before a user can be disconnected from a group he must ensure that it is not his current default group. If it is, the default must be set to some other group the user is connected to, say DRCS. The command to achieve this would be -

```
DEFGROUP DRCS
```

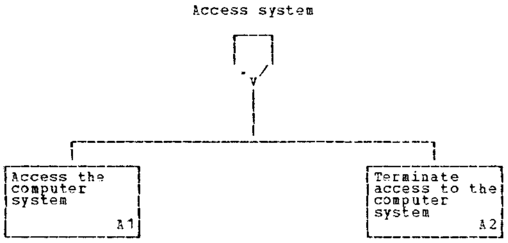
APPENDIX IV

DESCRIPTIONS AND HIPO CHARTS OF COMMANDS AND EXITS

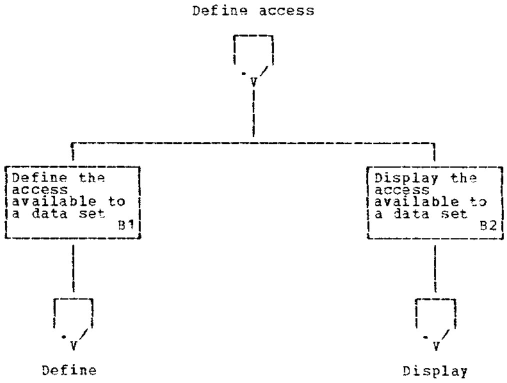
Table of contents of HIPO charts describing the operation of PACF to control access to system resources



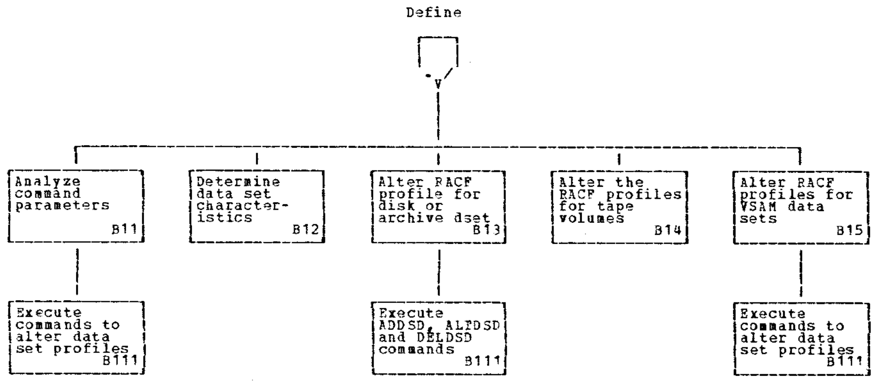
Control the access to the computer system


Access system
Access the computer system A1
Terminate access to the computer system A2

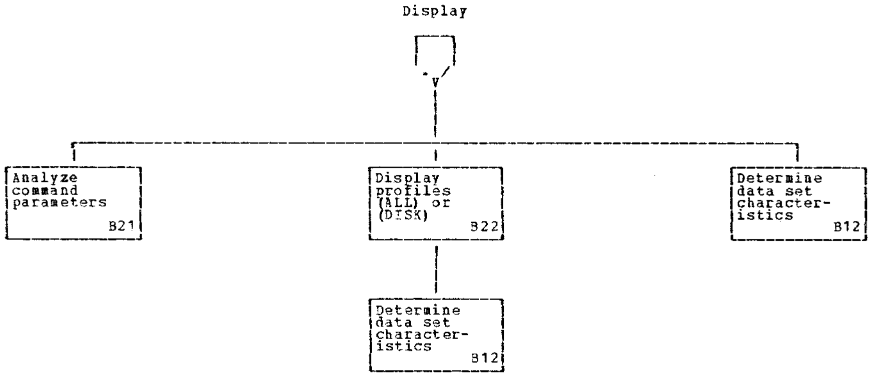
Define the access available to data sets


Define access
Define the access available to a data set B1
Display the access available to a data set B2
Define
Display

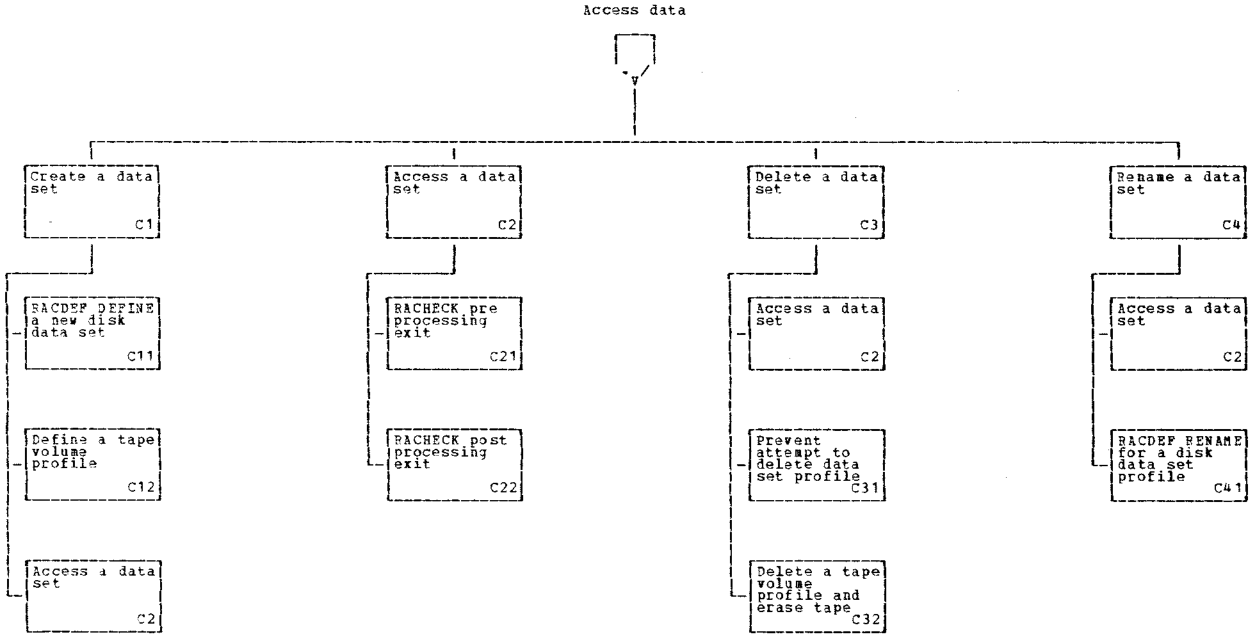
Define the access available to a data set


Define
Analyze command parameters B11
Determine data set character-istics B12
Alter RACF profile for disk or archive dset B13
Alter the RACF profiles for tape volumes B14
Alter RACF profiles for VSAM data sets B15
Execute commands to alter data set profiles B111
Execute ADDSD, ALTDSD and DELDSD commands B111
Execute commands to alter data set profiles B111

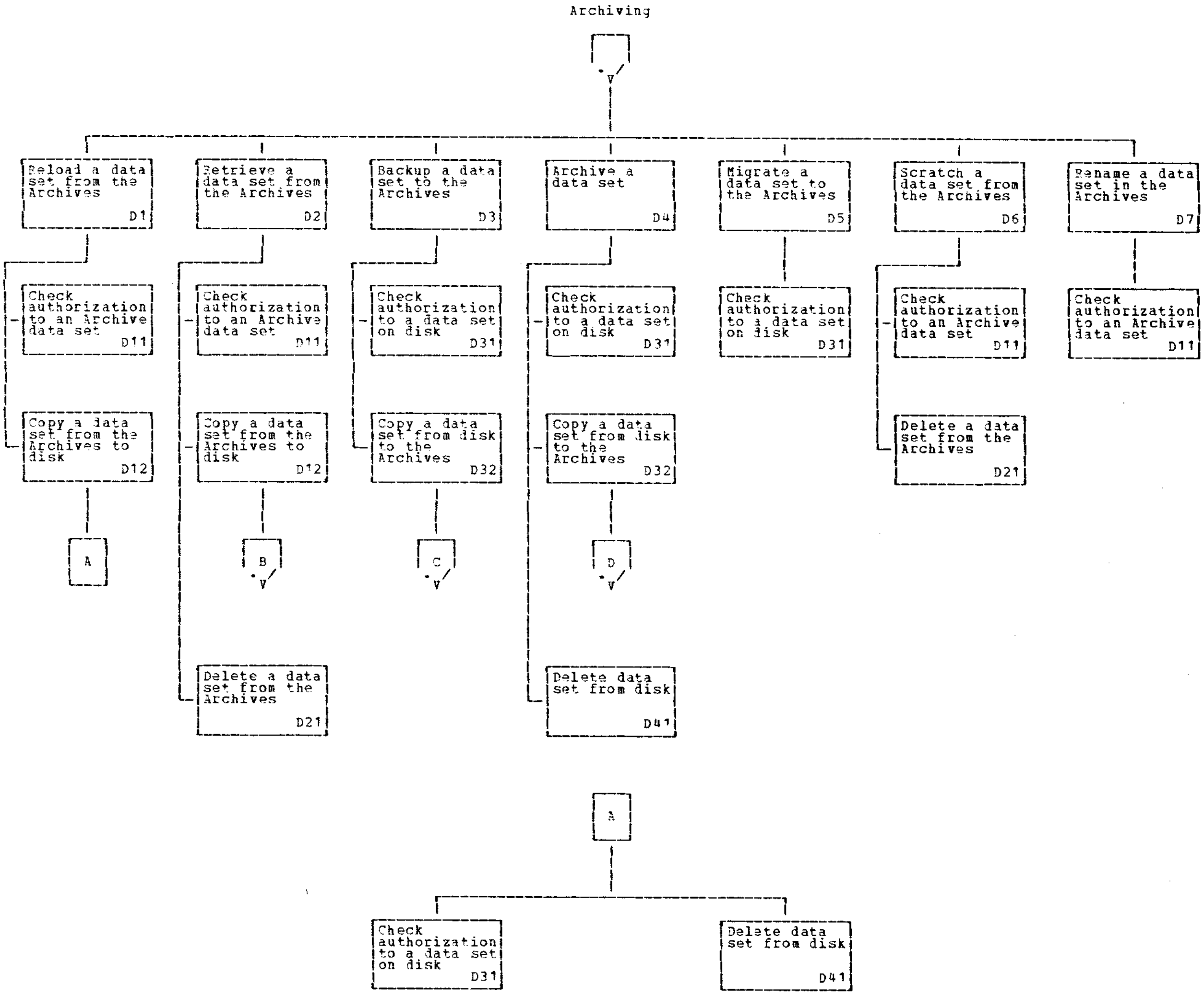
Display the access available to a data set


Display
Analyze command parameters B21
Display profiles (ALL) or (DISK) B22
Determine data set character-istics B12
Determine data set character-istics B12

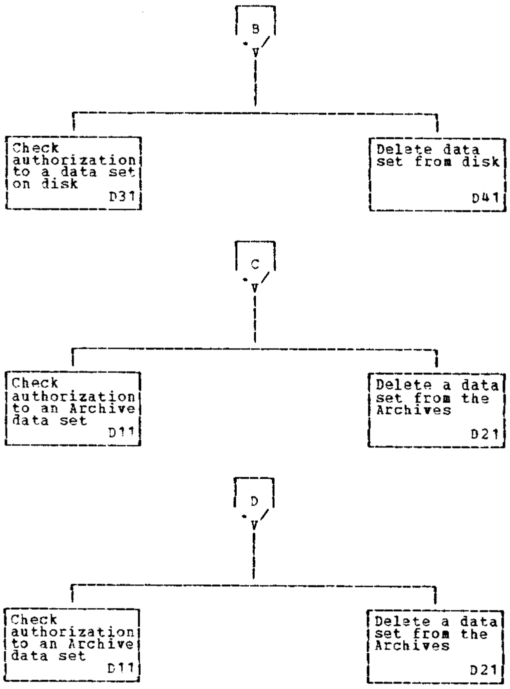
Control the access to data sets

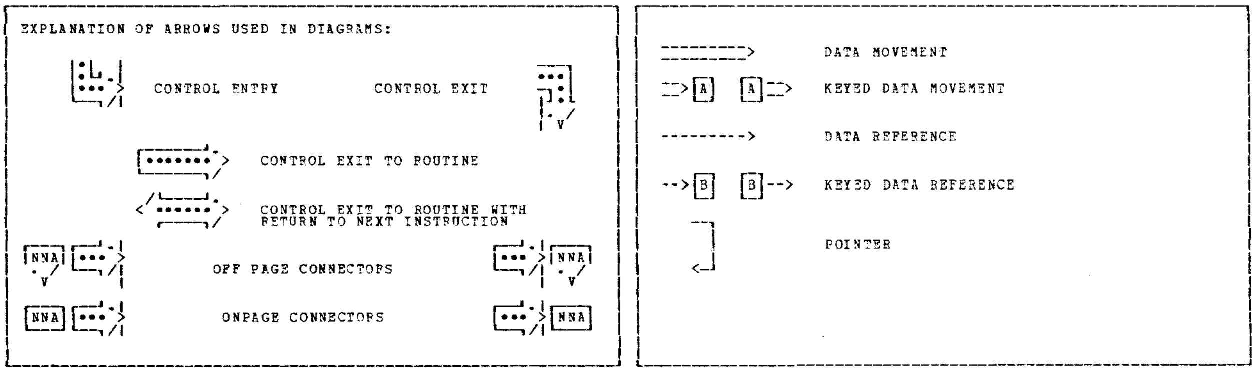

Access data
Create a data set C1
Access a data set C2
Delete a data set C3
Rename a data set C4
RACDEF DEFINE a new disk data set C11
Define a tape volume profile C12
Access a data set C2
RACHECK pre processing exit C21
RACHECK post processing exit C22
Access a data set C2
Prevent attempt to delete data set profile C31
Delete a tape volume profile and erase tape C32
Access a data set C2
RACDEF RENAME for a disk data set profile C41

Control the access to Archive data sets

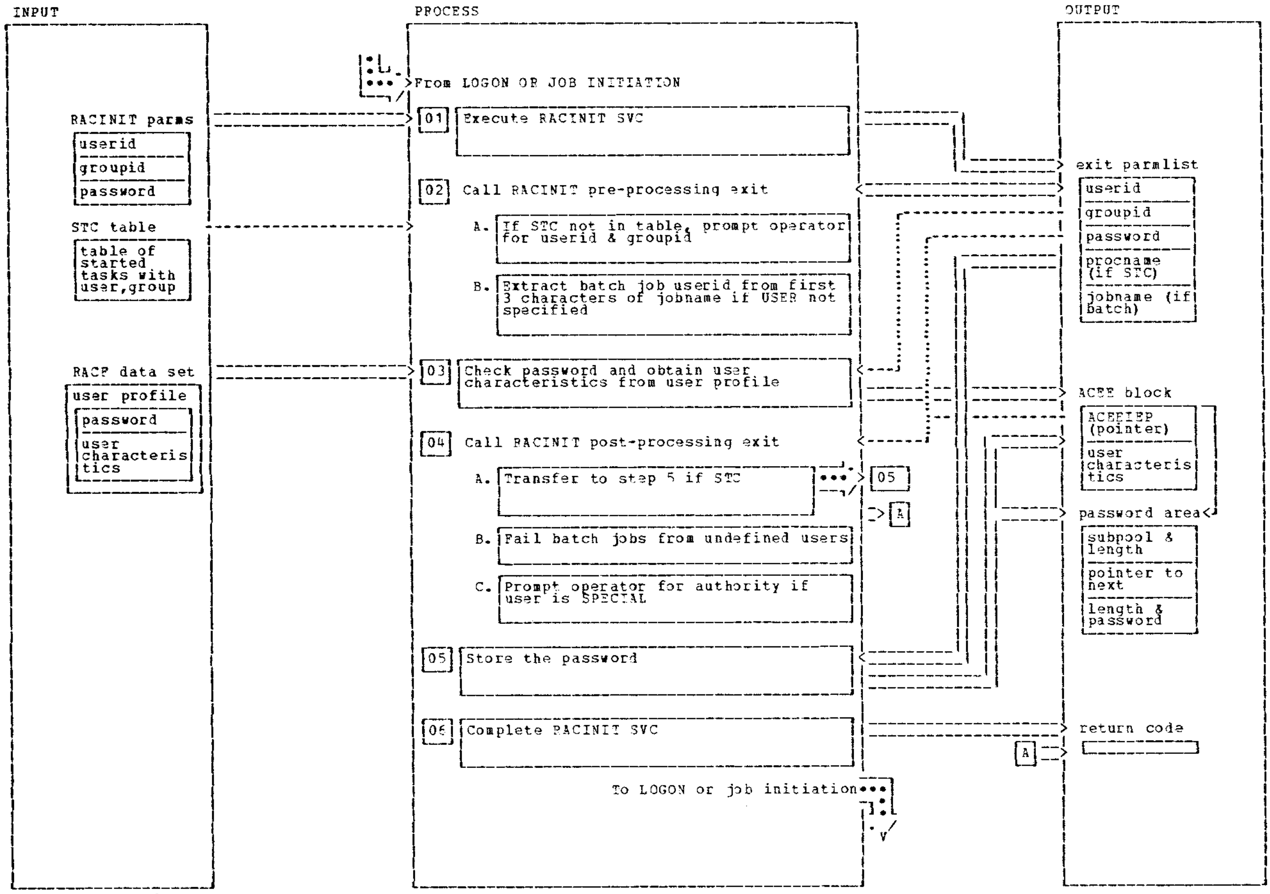

Archiving
Reload a data set from the Archives D1
Retrieve a data set from the Archives D2
Backup a data set to the Archives D3
Archive a data set D4
Migrate a data set to the Archives D5
Scratch a data set from the Archives D6
Rename a data set in the Archives D7
Check authorization to an Archive data set D11
Check authorization to a data set on disk D31
Copy a data set from the Archives to disk D12
Copy a data set from disk to the Archives D32
Delete a data set from the Archives D21
Delete data set from disk D41
A
B
C
D

Control the access to Archive data sets


B
Check authorization to a data set on disk D31
Delete data set from disk D41
C
Check authorization to an Archive data set D11
Delete a data set from the Archives D21
D
Check authorization to an Archive data set D11
Delete a data set from the Archives D21



Access the computer system


INPUT
PROCESS
OUTPUT
From LOGON OR JOB INITIATION
RACINIT parms
userid
groupid
password
STC table
table of started tasks with user,group
RACF data set
user profile
password
user characteristics
01
Execute RACINIT SVC
02
Call RACINIT pre-processing exit
A.
If STC not in table, prompt operator for userid & groupid
B.
Extract batch job userid from first 3 characters of jobname if USER not specified
03
Check password and obtain user characteristics from user profile
04
Call RACINIT post-processing exit
A.
Transfer to step 5 if STC
05
A
B.
Fail batch jobs from undefined users
C.
Prompt operator for authority if user is SPECIAL
05
Store the password
06
Complete RACINIT SVC
A
To LOGON or job initiation
exit parmlist
userid
groupid
password
procname (if STC)
jobname (if batch)
ACEE block
ACEEIEP (pointer)
user characteristics
password area
subpool & length
pointer to next
length & password
return code

HIPO-DIAGRAM A1

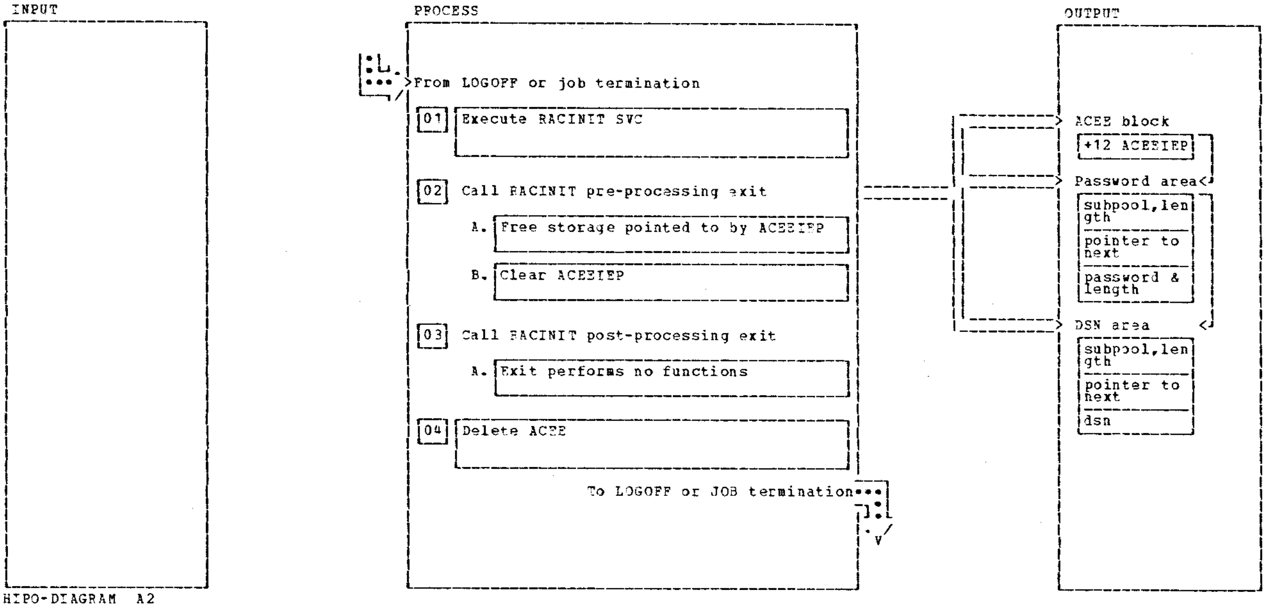
Access the computer system

NOTES	MODULE	LABEL	REF
1 RACINIT with the CREATE parameter.	ICHRIN01		
2A The STC table (ICHRIN03) contains an installation defined list of started tasks and their associated userid and groupid.	ICHRIX01		
2B At this installation, 3 character userids are coded as the first 3 characters of a batch job name.			
3 The RACINIT SVC checks the user profile.	ICHRIN01		
4B Certain users in this installation are not permitted to submit batch jobs or to LOGON except at specified secure terminals since their passwords are not private. In these cases, the exit has to cause the RACINIT to be repeated since only the pre-processing exit has a return code to cause RACINIT to fail	ICHRIX02		

NOTES	MODULE	LABEL	REF
4C SPECIAL users in this installation may not LOGON or submit jobs without operator authority.			
5 A storage area is obtained and the password stored so that batch jobs can submit other jobs to the internal reader. The ACEEIEP (installation word) is set to point to the password area.			

HIPO-DIAGRAM A1

Terminate access to the computer system



INPUT

PROCESS

From LOGOFF or job termination

01 Execute RACINIT SVC

02 Call RACINIT pre-processing exit

A. Free storage pointed to by ACEEIEP

B. Clear ACEEIEP

03 Call RACINIT post-processing exit

A. Exit performs no functions

04 Delete ACEE

To LOGOFF or JOB termination

OUTPUT

ACEE block

+12 ACEEIEP

Password area

subpool,length

pointer to next

password & length

DSN area

subpool,length

pointer to next

dsn

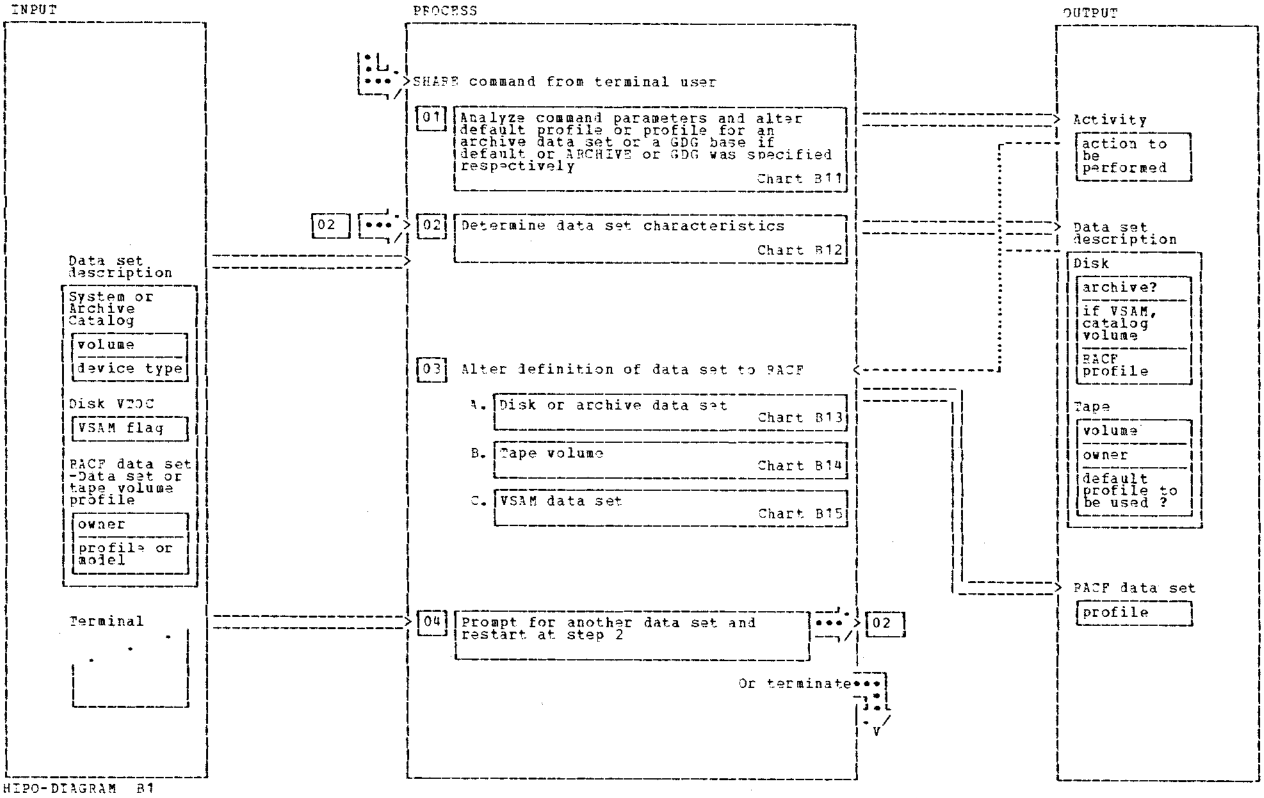
HIPO-DIAGRAM A2

Terminate access to the computer system

NOTES	MODULE	LABEL	REF
1 PACINIT with the DELETE parameter.			
2 Storage for the password and a list of data set names is obtained by exits during RACINIT CREATE and RACHECK respectively. The ACEEIEP (installation word) points to the chain of storage.	ICHRIX01		

HIPO-DIAGRAM A2

Define the access available to a data set - SHARE command



INPUT

Data set description

System or Archive Catalog

volume

device type

Disk VTOC

VSAM flag

RACF data set -Data set or tape volume profile

owner

profile or model

Terminal

PROCESS

SHARE command from terminal user

01 Analyze command parameters and alter default profile or profile for an archive data set or a GDG base if default or ARCHIVE or GDG was specified respectively Chart B11

02 02 Determine data set characteristics Chart B12

03 Alter definition of data set to RACF

A. Disk or archive data set Chart B13

B. Tape volume Chart B14

C. VSAM data set Chart B15

04 Prompt for another data set and restart at step 2 02

Or terminate

OUTPUT

Activity

action to be performed

Data set description

Disk

archive?

if VSAM, catalog volume

RACF profile

Tape

volume

owner

default profile to be used ?

RACF data set

profile

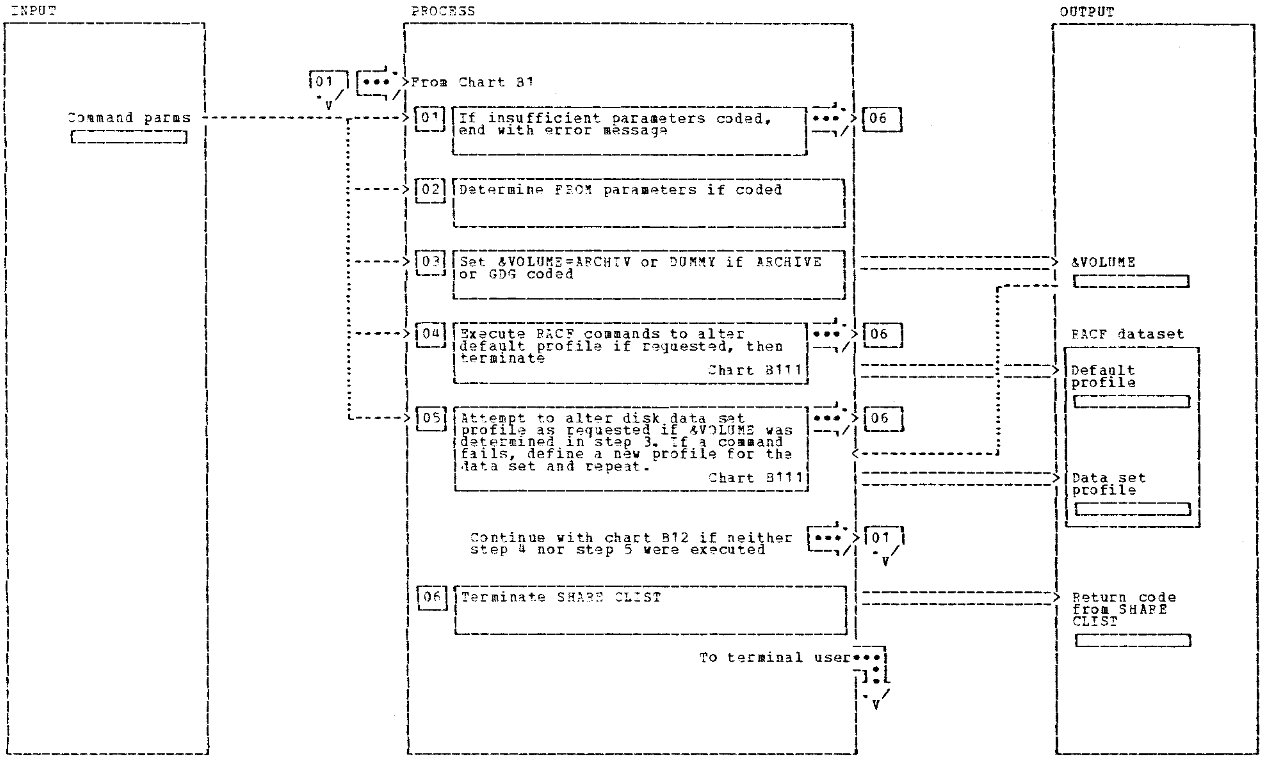
HIPO-DIAGRAM B1

Define the access available to a data set - SHARE command

NOTES	MODULE	LABEL	REF
1 The parameters specified by the user are analyzed to determine which RACF profile is to be altered, created or deleted, and what alterations are to be made to the profiles. If DEFAULT, ARCHIVE or GDG was specified then the profile can immediately be altered, since no further information is needed.	LISTP		
2 Execute the CATFIND command which creates a CLIST to be executed by the calling CLIST to obtain information derived by CATFIND.			
3 Execute the appropriate RACF commands to make the desired changes in the RACF profile for the disk data set or tape volume.			
4 Prompting for additional data sets can be requested by a parameter of the SHARE CLIST.			

HIPO-DIAGRAM B1

Analyze SHARE command parameters and modify default, ARCHIVE or GDG data set profile


INPUT
PROCESS
OUTPUT
Command parms
01
From Chart B1
01 If insufficient parameters coded, end with error message
06
02 Determine FROM parameters if coded
03 Set &VOLUME=ARCHIV or DUMMY if ARCHIVE or GDG coded
&VOLUME
04 Execute RACF commands to alter default profile if requested, then terminate
Chart B111
06
RACF dataset
Default profile
05 Attempt to alter disk data set profile as requested if &VOLUME was determined in step 3. If a command fails, define a new profile for the data set and repeat.
Chart B111
06
Data set profile
Continue with chart B12 if neither step 4 nor step 5 were executed
01
06 Terminate SHARE CLIST
Return code from SHARE CLIST
To terminal user

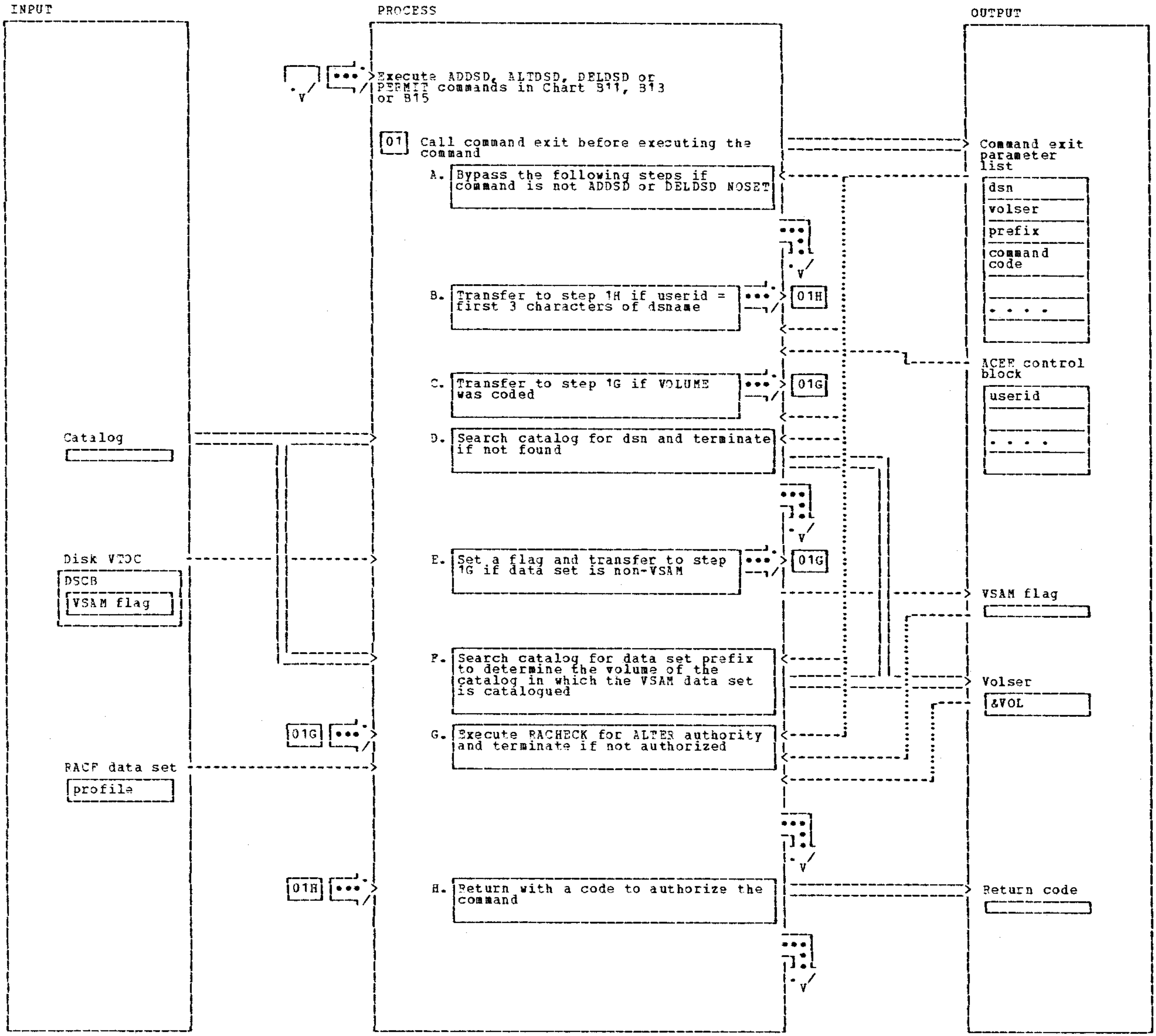
HIPO-DIAGRAM B11

Analyze SHARE command parameters and modify default, ARCHIVE or GDG data set profile

NOTES	MODULE	LABEL	REF
1 Error if Command parameters are not sufficient to cause any change to a RACF profile.			
2 The type of FROM dataset is determined and the FROM and FCLASS parameters are set up for use in a PERMIT command in step 4 or 5.			
3 ARCHIVE only has to be coded if a data set of the same name is catalogued on tape or disk and the one in the archives is being referred to. GDG is coded for a disk GDG base name - all generations are SHAREd in the same way.			
4 Prevent the user from specifying UACC for the default profile since it is banned in this installation.			
5 If &VOLUME has been set, then it must be a disk data set profile - either for archives or GDG base. The normal reason for a command (ALTDSD or PERMIT) to fail is that a profile does not exist - i.e. the default applied. In this case an ADDSD NOSET command must first be issued to create the profile. This will fail if the user does not have ALTER access in the default profile.			
6 The request is complete if step 4 or 5 was executed.			

HIPO-DIAGRAM B11

Execute ADDSD, ALTDSD and DELDSD commands


INPUT
PROCESS
OUTPUT
Execute ADDSD, ALTDSD, DELDSD or PERMIT commands in Chart B11, B13 or B15
01 Call command exit before executing the command
A. Bypass the following steps if command is not ADDSD or DELDSD NOSET
B. Transfer to step 1H if userid = first 3 characters of dsname
01H
C. Transfer to step 1G if VOLUME was coded
01G
D. Search catalog for dsn and terminate if not found
E. Set a flag and transfer to step 1G if data set is non-VSAM
01G
F. Search catalog for data set prefix to determine the volume of the catalog in which the VSAM data set is catalogued
01G
G. Execute RACHECK for ALTER authority and terminate if not authorized
01H
H. Return with a code to authorize the command
Catalog
Disk VTOC
DSCB
VSAM flag
RACF data set
profile
Command exit parameter list
dsn
volser
prefix
command code
. . . .
ACEE control block
userid
. . . .
VSAM flag
Volser
&VOL
Return code

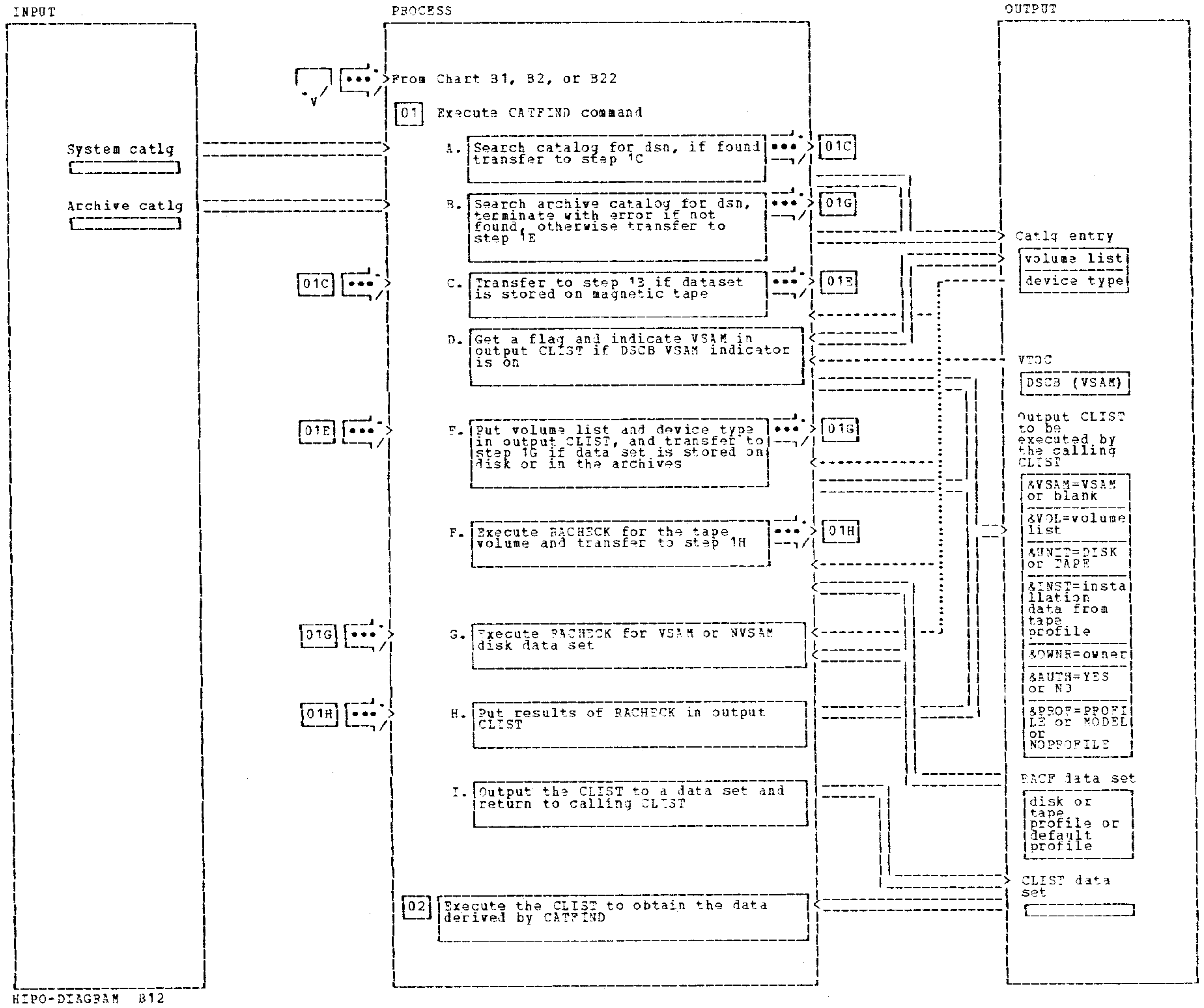
HIPO-DIAGRAM B111

Execute ADDSD, ALTDSD and DELDSD commands

	NOTES	MODULE	LABEL	REF
1	The RACF command processor calls the installation coded command exit before executing the RACF command	ICHCNX00		
1A	The normal checks on commands other than the NOSET commands are satisfactory - NOSET commands are only allowed for the data set owner in standard RACF.			
1B	In this installation userids are 3 characters but data sets owned by a user may have more than 3 characters in the prefix as long as the first 3 equal the userid.			
1F	RACHECK for a VSAM data set requires the volume of the catalog to be specified.			

HIPO-DIAGRAM B111

Determine data set characteristics



From Chart B1, B2, or B22

01 Execute CATFIND command

A. Search catalog for dsn, if found transfer to step 1C

B. Search archive catalog for dsn, terminate with error if not found, otherwise transfer to step 1E

C. Transfer to step 1E if dataset is stored on magnetic tape

D. Get a flag and indicate VSAM in output CLIST if DSCB VSAM indicator is on

E. Put volume list and device type in output CLIST, and transfer to step 1G if data set is stored on disk or in the archives

F. Execute RACHECK for the tape volume and transfer to step 1H

G. Execute RACHECK for VSAM or NVSAM disk data set

H. Put results of RACHECK in output CLIST

I. Output the CLIST to a data set and return to calling CLIST

02 Execute the CLIST to obtain the data derived by CATFIND

HIPO-DIAGRAM B12

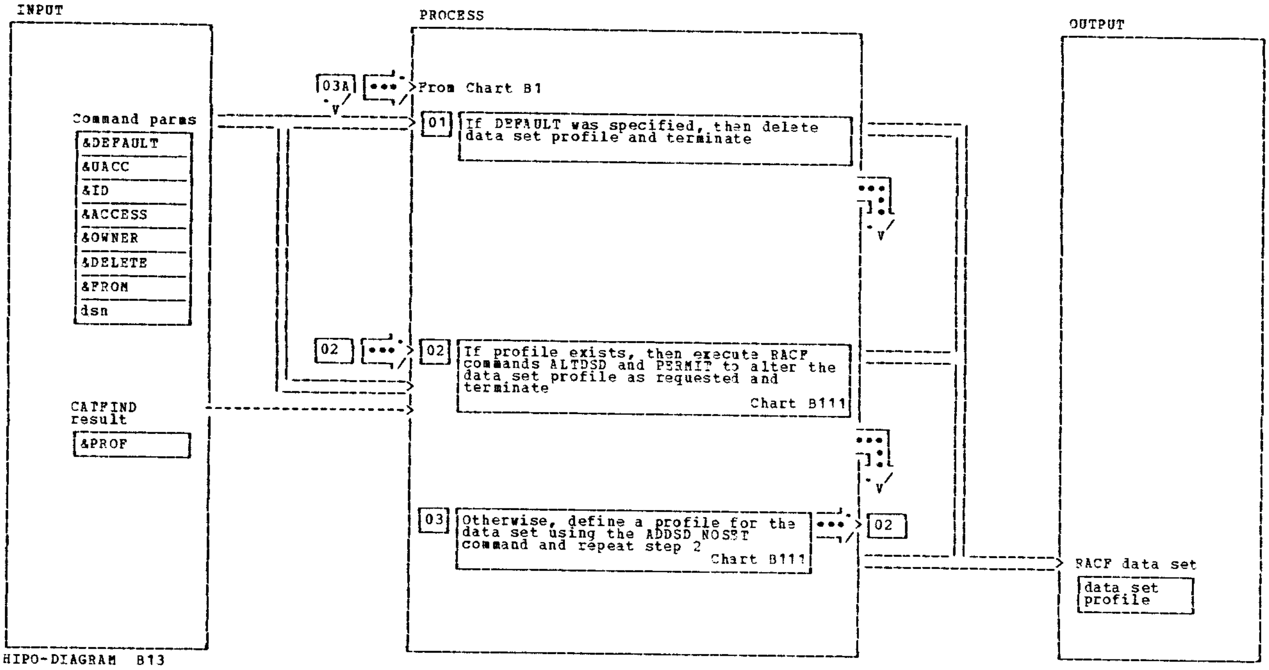
Determine data set characteristics

NOTES	MODULF	LABEL	FEF
1 The SHARE or LISTP CLISTs execute the CATFIND command to determine the characteristics of the data set.			
1A The volume list and unit type are obtained.	CATFIND		
1B The data set may be in the archives if it is not catalogued.			
1D If data set is VSAM, search the catalog for the data set prefix - the volume of the catalog in which the data set is catalogued is obtained.			

NOTES	MODULE	LABEL	REF
1F The RACHECK macro is executed with the CSA option which causes a copy of the profile to be placed in storage so that the command may access fields in the profile.			
1H The CATFIND command creates a CLIST which the calling CLIST may execute to obtain the results of the CATFIND command.			

HIPO-DIAGRAM B12

Alter RACF profile for disk or archive data set


INPUT
PROCESS
OUTPUT
03A
From Chart B1
Command parms
&DEFAULT
&UACC
&ID
&ACCESS
&OWNER
&DELETE
&FROM
dsn
01
If DEFAULT was specified, then delete data set profile and terminate
02
02
If profile exists, then execute RACF commands ALTDSD and PERMIT to alter the data set profile as requested and terminate
Chart B111
CATFIND result
&PROF
03
Otherwise, define a profile for the data set using the ADDSD NOSET command and repeat step 2
Chart B111
02
RACF data set
data set profile

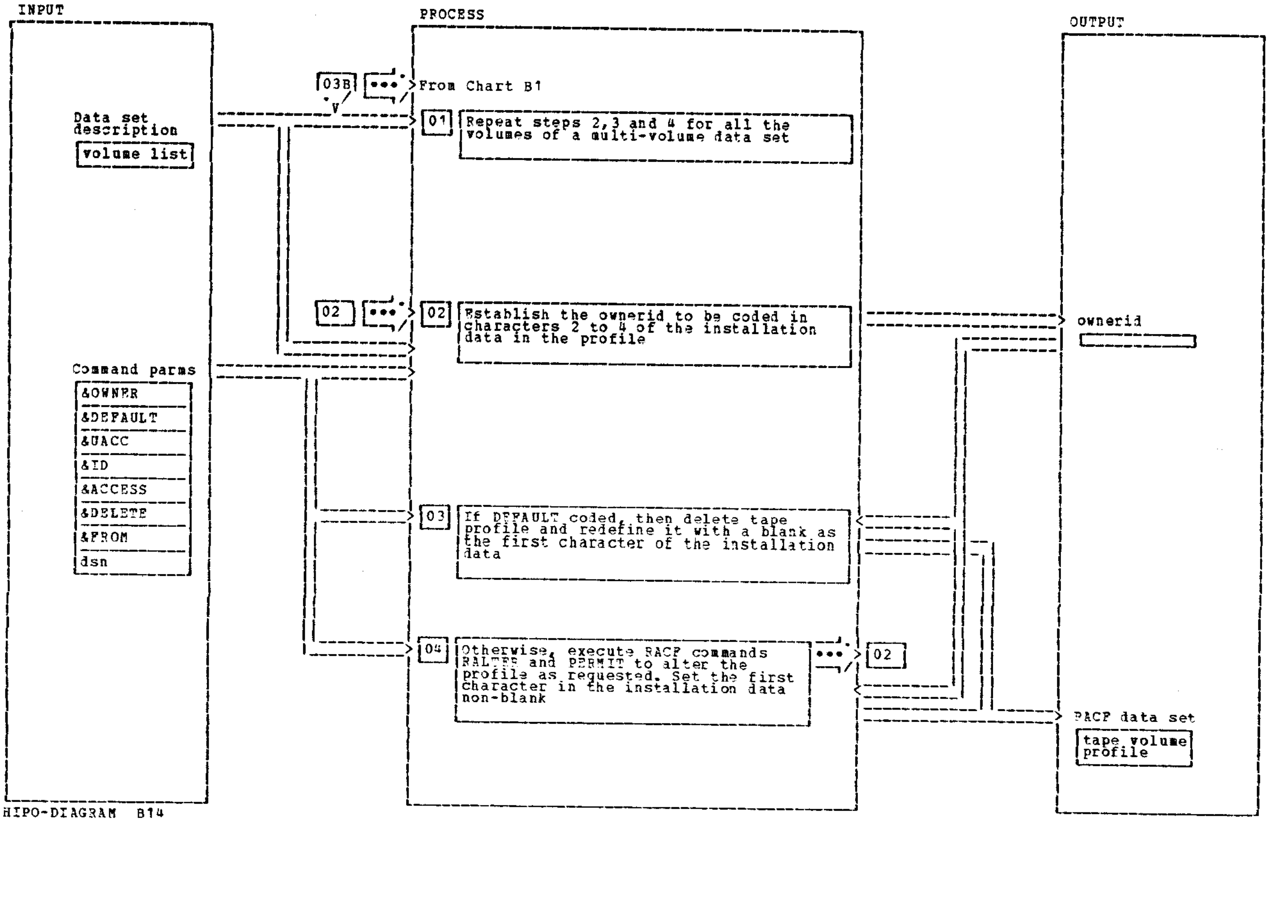
HIPO-DIAGRAM B13

Alter RACF profile for disk or archive data set

NOTES	MODULE	LABEL	REF	NOTES	MODULE	LABEL	REF
RACF normally only allows the DELDSD and ADDSD commands with the NOSET parameter to be executed by the user whose userid prefixes a data set, or a SPECIAL user. However in this installation a command exit executes a RACHECK to determine whether the user has ALTER access authority and if so, authorizes these commands. In the case of ADDSD the user must have ALTER authority in the default profile.							

HIPO-DIAGRAM B13

Alter the RACF profile for tape volumes


INPUT
PROCESS
OUTPUT
03B
From Chart B1
Data set description
volume list
01
Repeat steps 2,3 and 4 for all the volumes of a multi-volume data set
02
02
Establish the ownerid to be coded in characters 2 to 4 of the installation data in the profile
ownerid
Command parms
&OWNER
&DEFAULT
&UACC
&ID
&ACCESS
&DELETE
&FROM
dsn
03
If DEFAULT coded, then delete tape profile and redefine it with a blank as the first character of the installation data
04
Otherwise, execute RACF commands RALTER and PERMIT to alter the profile as requested. Set the first character in the installation data non-blank
02
RACF data set
tape volume profile

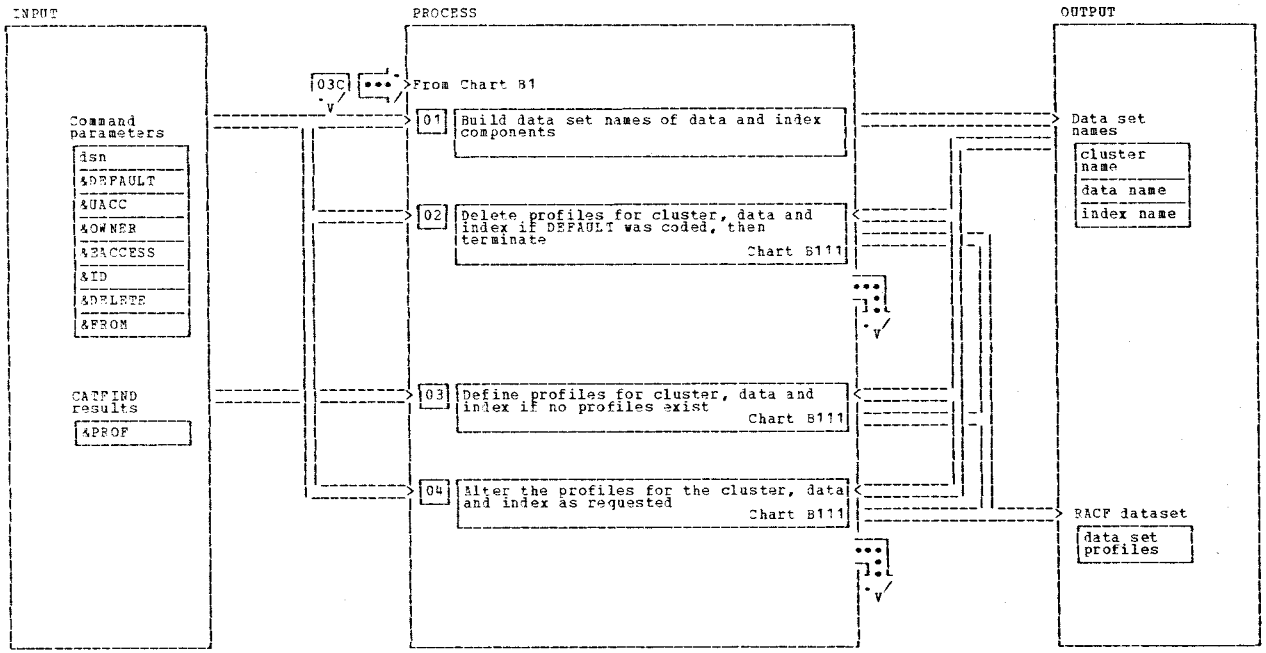
HIPO-DIAGRAM B14

Alter the RACF profile for tape volumes

NOTES	MODULE	LABEL	REF	NOTES	MODULE	LABEL	REF
2 The ownerid in the installation data is the first 3 characters of the data set name. If the first character of the installation data is blank it indicates that the default profile should be used to determine the access available to the data set. In this case, the data set name is not available to the RACHECK exits to determine whose default profile should be used. Therefore the first 3 characters of the data set name are also stored in the installation data (all default profiles have 3 character prefixes).				3 In this installation all tape volumes which contain a catalogued data set will have a profile, except momentarily when the SHARE CLIST deletes a profile preparatory to redefining it to indicate the default. Therefore it is safe to allow all users to define a profile for any tape volume.			

HIPO-DIAGRAM B14

Alter RACF profiles for VSAM data sets


INPUT
PROCESS
OUTPUT
Command parameters
dsn
&DEFAULT
&UACC
&OWNER
&BACCESS
&ID
&DELETE
&FROM
CATFIND results
&PROF
03C
•••
From Chart B1
01 Build data set names of data and index components
02 Delete profiles for cluster, data and index if DEFAULT was coded, then terminate
Chart B111
•••
03 Define profiles for cluster, data and index if no profiles exist
Chart B111
04 Alter the profiles for the cluster, data and index as requested
Chart B111
•••
Data set names
cluster name
data name
index name
RACF dataset
data set profiles

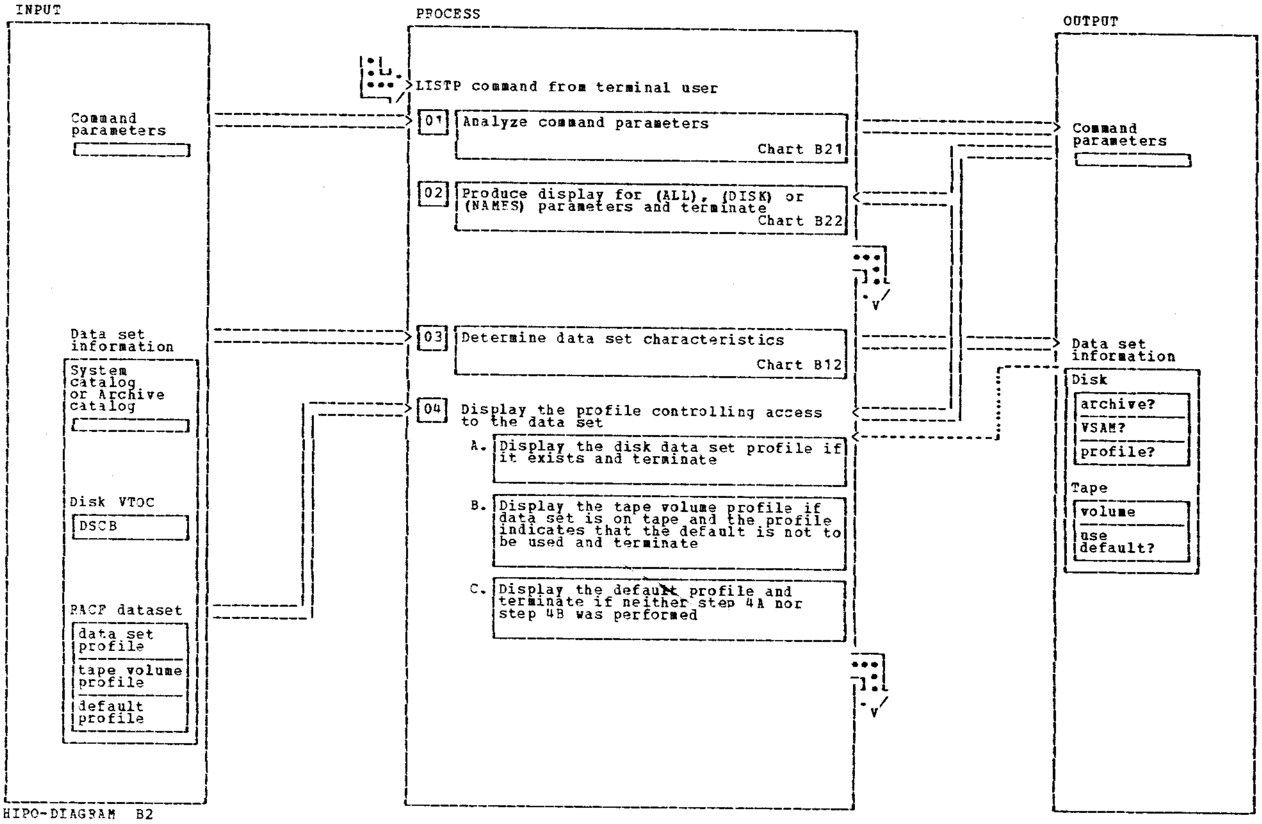
HIPO-DIAGRAM B15

Alter RACF profiles for VSAM data sets

NOTES	MODULE	LABEL	REF	NOTES	MODULE	LABEL	REF
1 In this installation, all VSAM data set names standardly have clustername.DATA and clustername.INDEX as the names of the data and index components respectively.				2 The default profile is used to define the access available to a disk data set if no RACF profile exists for the data set. 3 The ADDSD command is used. 4 The ALTDSD and PERMIT commands are used.			

HIPO-DIAGRAM B15

Display the access available to a data set


INPUT
PROCESS
OUTPUT
LISTP command from terminal user
Command parameters
01 Analyze command parameters
Chart B21
Command parameters
02 Produce display for (ALL), (DISK) or (NAMES) parameters and terminate
Chart B22
Data set information
System catalog or Archive catalog
03 Determine data set characteristics
Chart B12
Data set information
Disk
archive?
VSAM?
profile?
Tape
volume
use default?
Disk VTOC
DSCB
04 Display the profile controlling access to the data set
A. Display the disk data set profile if it exists and terminate
B. Display the tape volume profile if data set is on tape and the profile indicates that the default is not to be used and terminate
C. Display the default profile and terminate if neither step 4A nor step 4B was performed
RACF dataset
data set profile
tape volume profile
default profile

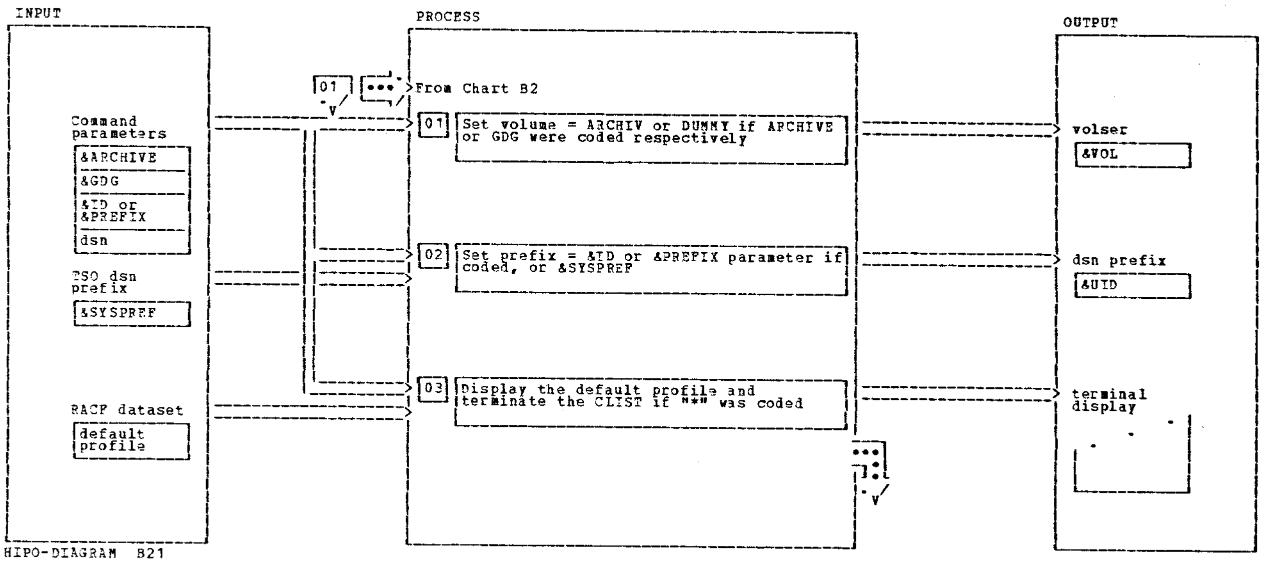
HIPO-DIAGRAM B2

Display the access available to a data set

	NOTES	MODULE	LABEL	REF
4A	The access to a disk data set is controlled by the owner's default profile unless a specific profile exists for the data set.			
4B	The access to a tape data set is controlled by the default profile unless the first character of the installation data in the tape profile is non-blank.			

HIPO-DIAGRAM B2

Analyze the parameters of the LISTP command


INPUT
PROCESS
OUTPUT
01
From Chart B2
Command parameters
&ARCHIVE
&GDG
&ID or &PREFIX
dsn
01 Set volume = ARCHIV or DUMMY if ARCHIVE or GDG were coded respectively
volser
&VOL
TSO dsn prefix
&SYSPREF
02 Set prefix = &ID or &PREFIX parameter if coded, or &SYSPREF
dsn prefix
&UID
RACF dataset
default profile
03 Display the default profile and terminate the CLIST if "*" was coded
terminal display

HIPO-DIAGRAM B21

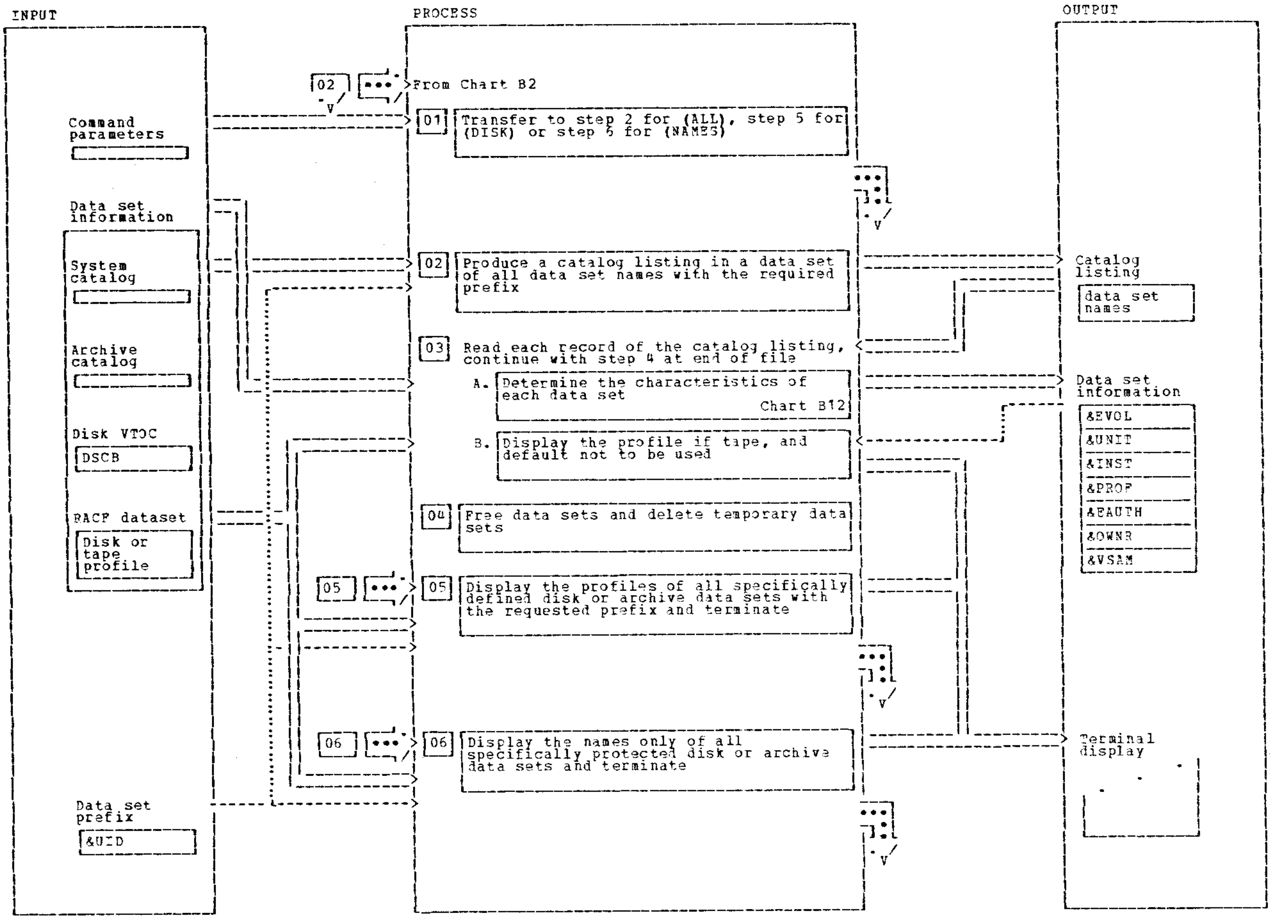
Analyze the parameters of the LISTP command

NOTES	MODULE	LABEL	REF
3 "*" coded as the dsn indicates display the default profile.			

NOTES	MODULE	LABEL	REF

HIPO-DIAGRAM B21

Produce displays for the LISTP parameters (ALL), (DISK) and (NAMES)



01 Transfer to step 2 for (ALL), step 5 for (DISK) or step 6 for (NAMES)

02 Produce a catalog listing in a data set of all data set names with the required prefix

03 Read each record of the catalog listing, continue with step 4 at end of file

A. Determine the characteristics of each data set Chart B12

B. Display the profile if tape, and default not to be used

04 Free data sets and delete temporary data sets

05 Display the profiles of all specifically defined disk or archive data sets with the requested prefix and terminate

06 Display the names only of all specifically protected disk or archive data sets and terminate

HIPO-DIAGRAM B22

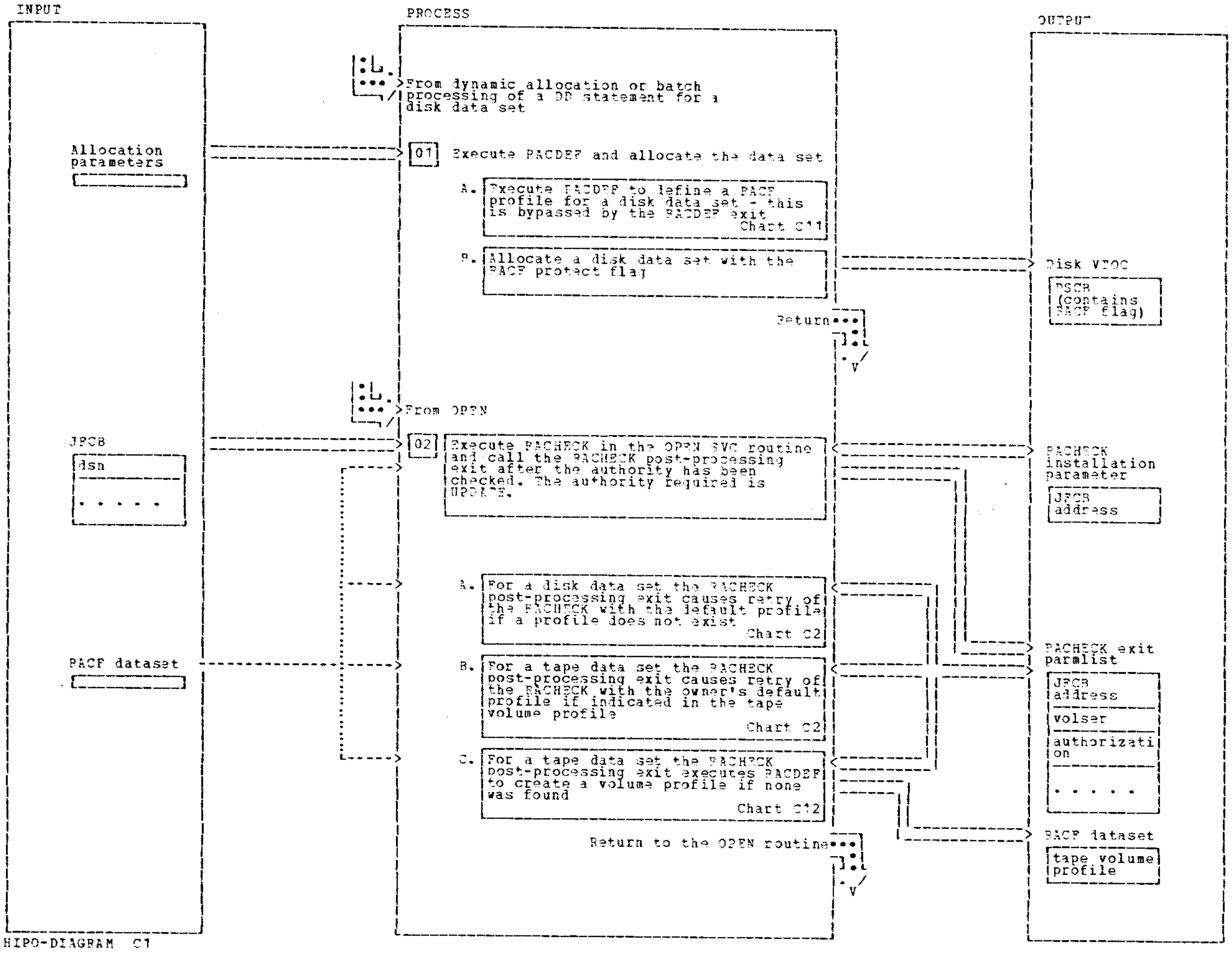
Produce displays for the LISTP parameters (ALL), (DISK) and (NAMES)

NOTES	MODULE	LABEL	REF
3B The default profile determines the access to a tape data set unless the first character of the installation data in the tape volume profile is non-blank.			
4 Several data sets are used during the above steps.			

NOTES	MODULE	LABEL	REF
5 A single RACF command can be used to display specifically defined disk data sets since profiles exist only for these but the complications of step 3 are necessary for tape since a profile exists for each tape volume.			
6 A RACF SEARCH command can be used since a profile only exists for each specifically defined data set.			

HIPO-DIAGRAM B22

Create a data set


INPUT
PROCESS
OUTPUT
Allocation parameters
JFCB
dsn
. . . . .
RACF dataset
From dynamic allocation or batch processing of a DD statement for a disk data set
01 Execute RACDEF and allocate the data set
A. Execute RACDEF to define a RACF profile for a disk data set - this is bypassed by the RACDEF exit Chart C11
B. Allocate a disk data set with the RACF protect flag
Return
From OPEN
02 Execute RACHECK in the OPEN SVC routine and call the RACHECK post-processing exit after the authority has been checked. The authority required is UPDATE.
A. For a disk data set the RACHECK post-processing exit causes retry of the RACHECK with the default profile if a profile does not exist Chart C2
B. For a tape data set the RACHECK post-processing exit causes retry of the RACHECK with the owner's default profile if indicated in the tape volume profile Chart C2
C. For a tape data set the RACHECK post-processing exit executes RACDEF to create a volume profile if none was found Chart C12
Return to the OPEN routine
Disk VTOC
DSCB (contains RACF flag)
RACHECK installation parameter
JFCB address
RACHECK exit parmlist
JFCB address
volser
authorization
. . . . .
RACF dataset
tape volume profile

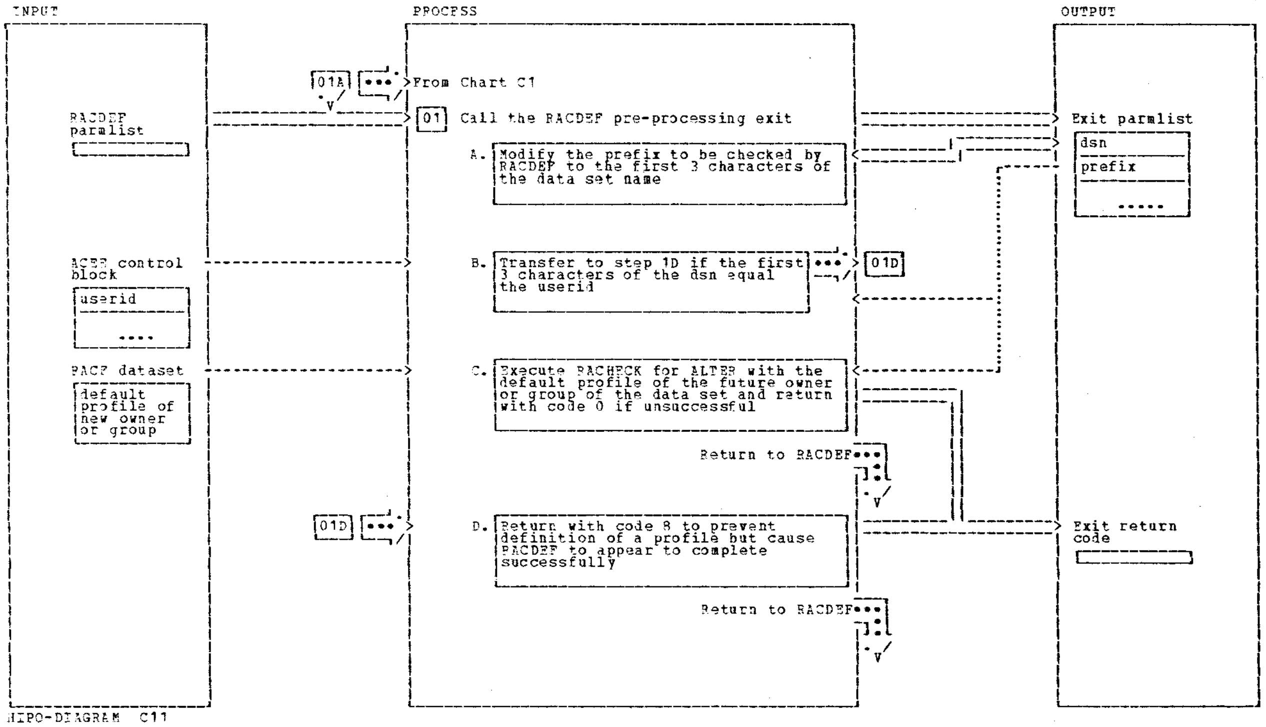
HIPO-DIAGRAM C1

Create a data set

	NOTES	MODULE	LABEL	REF
1A	In this installation, access to a disk data set is controlled by a default profile unless a profile is specifically defined for the data set. Therefore a profile is not created when a data set is created.			
1B	All users are given ADSP so all data sets are automatically protected when created.			
2	The OPEN routines have been modified to pass the JFCB as an installation parameter to RACHECK and thence to the RACHECK exits in the case of a new tape data set. The data set name prefix is needed to establish the ownership of the tape in the case where a user creates a tape data set not his own. (The JFCB contains the data set name).			
2A	Since the data set is just being created it will not have a specific profile - see step 01A above.			
2B	If the tape volume already contains one or more data sets it will have a profile.			
2C	If the tape has come from the scratch pool it will not have a profile, and one must be created when the first data set is written to it. This is done by executing a RACDEF in the RACHECK post-processing exit.			

HIPO-DIAGRAM C1

RACDEF DEFINE a new disk data set


INPUT
PROCESS
OUTPUT
RACDEF parmlist
ACEE control block
userid
....
RACF dataset
default profile of new owner or group
01A
From Chart C1
01
Call the RACDEF pre-processing exit
A. Modify the prefix to be checked by RACDEF to the first 3 characters of the data set name
B. Transfer to step 1D if the first 3 characters of the dsn equal the userid
01D
C. Execute RACHECK for ALTER with the default profile of the future owner or group of the data set and return with code 0 if unsuccessful
Return to RACDEF
01D
D. Return with code 8 to prevent definition of a profile but cause RACDEF to appear to complete successfully
Return to RACDEF
Exit parmlist
dsn
prefix
.....
Exit return code

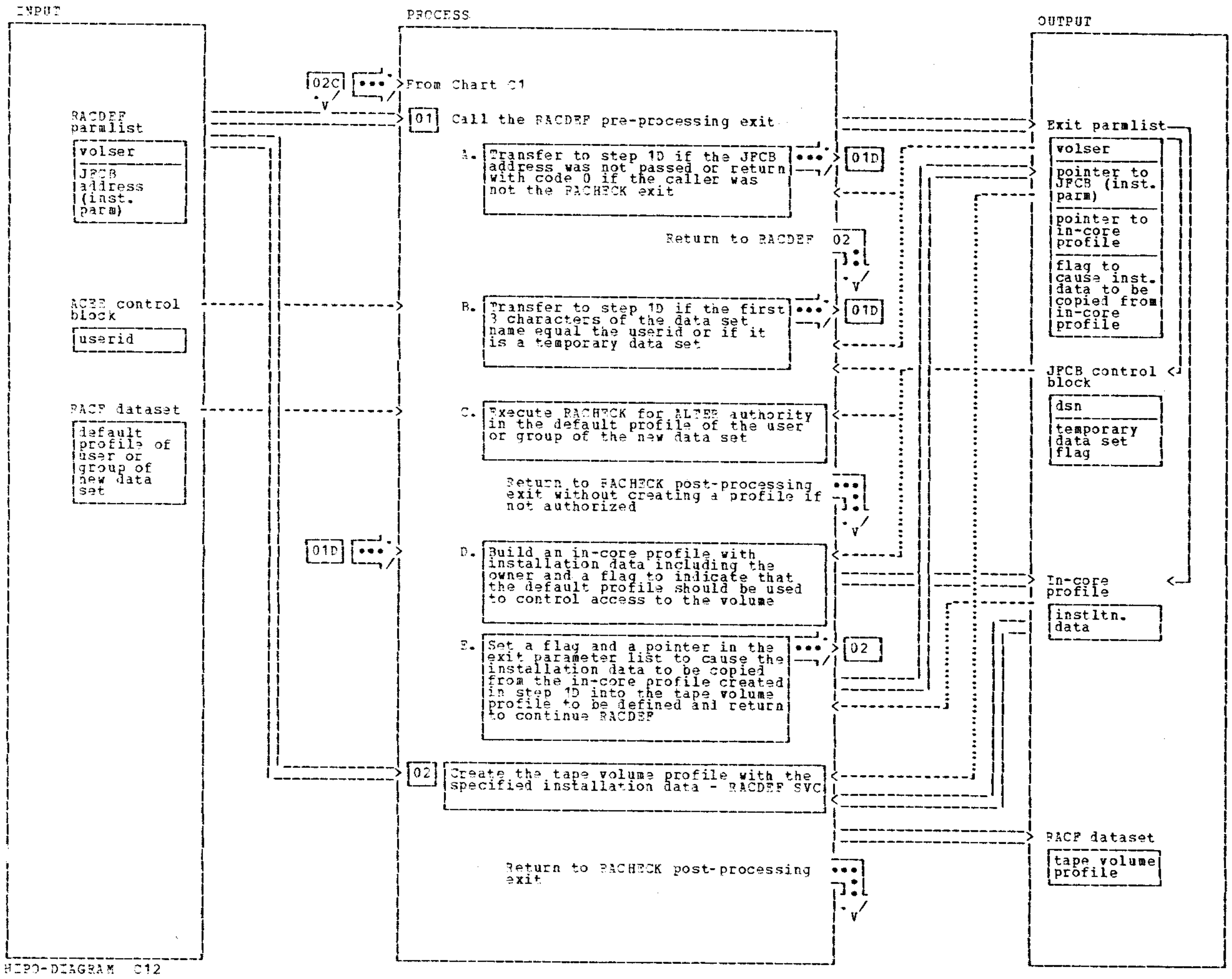
HIPO-DIAGRAM C11

RACDEF DEFINE a new disk data set

	NOTES	MODULE	LABEL	REF
1A	All userids and groupids in this installation are 3 characters and users may own data sets with a longer prefix as long as the first 3 characters equal their userids.	ICHRDX01		
1C	A data set may be created for another user only if ALTER authority is available in the other user's default profile. A return code of 0 from the exit causes RACDEF to continue normally in which case RACDEF will fail the request since RACF normally does not allow users to create data sets for others.			
1D	Data sets in this installation only have profiles if defined specifically. Access is normally controlled by a default profile for each user or group.			

HIPO-DIAGRAM C11

Define a tape volume profile



From Chart C1

01 Call the RACDEF pre-processing exit

A. Transfer to step 1D if the JFCB address was not passed or return with code 0 if the caller was not the RACHECK exit

Return to RACDEF

B. Transfer to step 1D if the first 3 characters of the data set name equal the userid or if it is a temporary data set

C. Execute RACHECK for ALTER authority in the default profile of the user or group of the new data set

Return to RACHECK post-processing exit without creating a profile if not authorized

D. Build an in-core profile with installation data including the owner and a flag to indicate that the default profile should be used to control access to the volume

E. Set a flag and a pointer in the exit parameter list to cause the installation data to be copied from the in-core profile created in step 1D into the tape volume profile to be defined and return to continue RACDEF

02 Create the tape volume profile with the specified installation data - RACDEF SVC

Return to RACHECK post-processing exit

HIPO-DIAGRAM C12

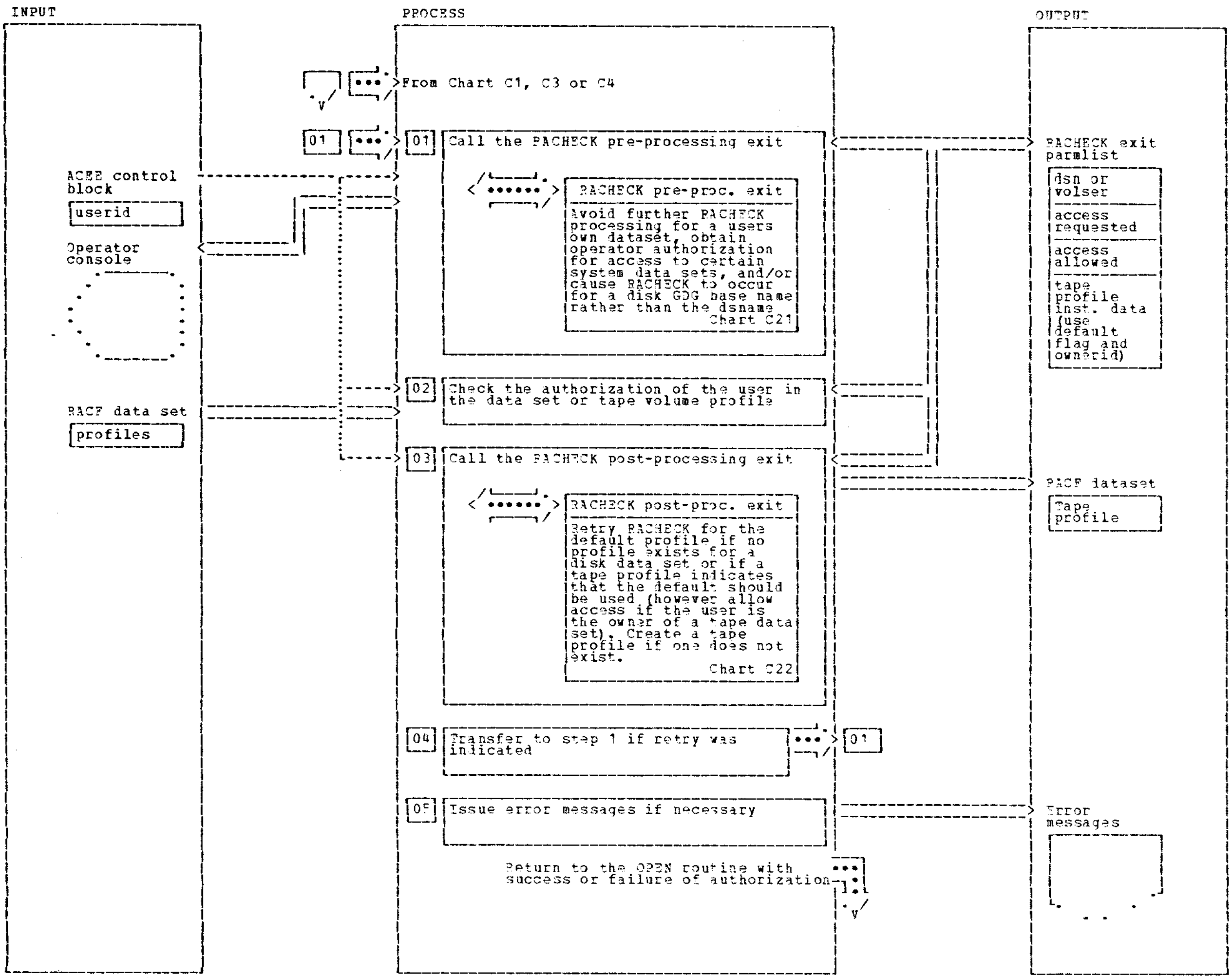
Define a tape volume profile

	NOTES	MODULE	LABEL	REF
1C	In this installation, a data set may be created for another user only if ALTER authority is available in the other user's default profile.			

	NOTES	MODULE	LABEL	REF
1E	When RACDEF subsequently creates the tape profile, it will copy the installation data from the in-core profile into the created profile. A pointer to the in-core profile and a flag in the exit parameters cause this to happen.			

HIPO-DIAGRAM C12

Access a data set



01 Call the RACHECK pre-processing exit

RACHECK pre-proc. exit

Avoid further RACHECK processing for a users own dataset, obtain operator authorization for access to certain system data sets, and/or cause RACHECK to occur for a disk GDG base name rather than the dsname
Chart C21

02 Check the authorization of the user in the data set or tape volume profile

03 Call the RACHECK post-processing exit

RACHECK post-proc. exit

Retry RACHECK for the default profile if no profile exists for a disk data set or if a tape profile indicates that the default should be used (however allow access if the user is the owner of a tape data set). Create a tape profile if one does not exist.
Chart C22

04 Transfer to step 1 if retry was indicated

05 Issue error messages if necessary

Return to the OPEN routine with success or failure of authorization

HIPO-DIAGRAM C2

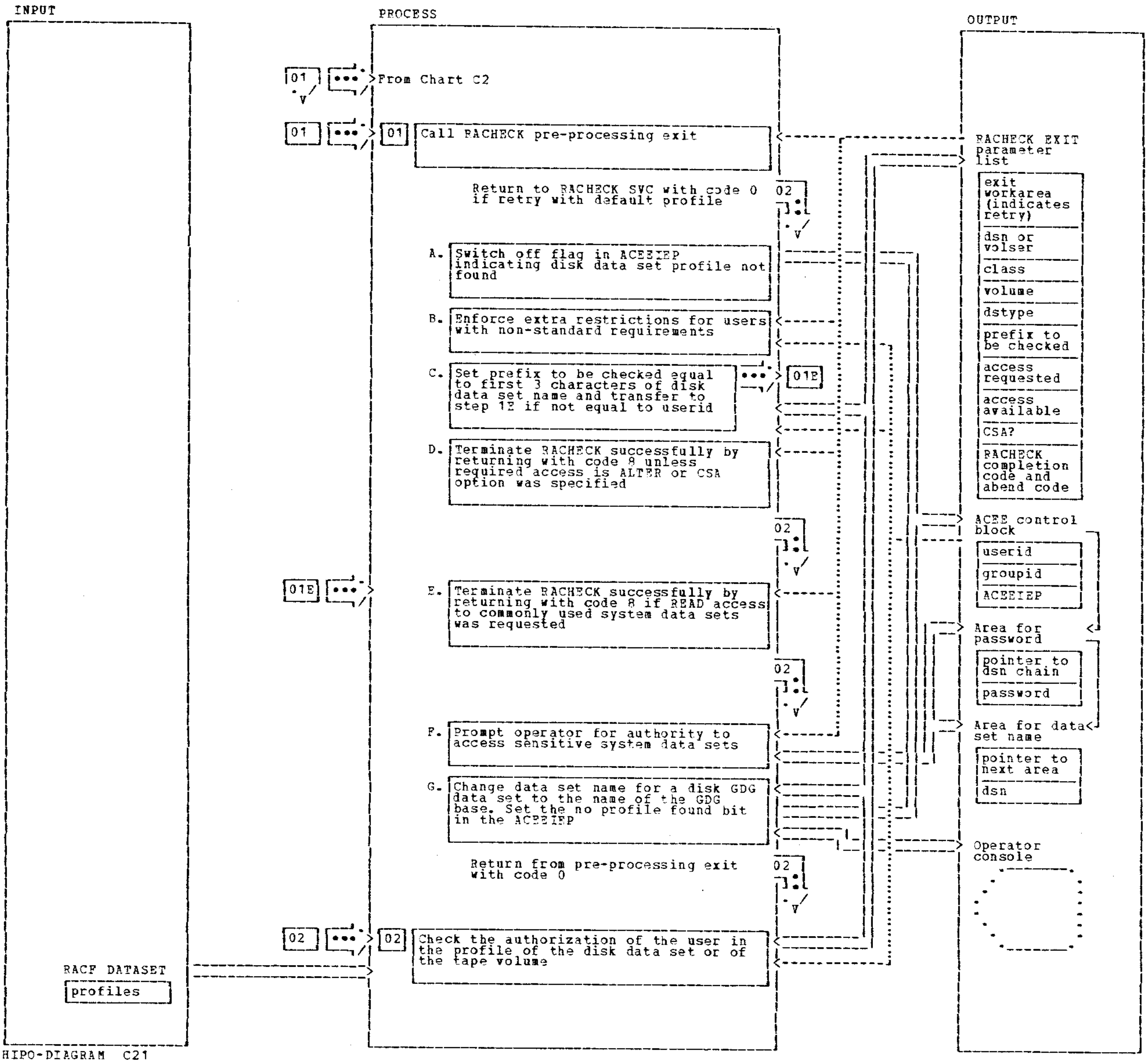
Access a data set

NOTES	MODULE	LABEL	REF
1 A return code from the exit can prevent further processing of RACHECK.	ICHRCX01		

NOTES	MODULE	LABEL	REF
3 A return code form the exit can cause the RACHECK to be repeated with the resource to be checked altered to the default profile. A flag has to be set to prevent loops.	ICHRCX02		

HIPO-DIAGRAM C2

Detailed actions of the RACHECK pre-processing exit for for data set access


INPUT

PROCESS

OUTPUT

From Chart C2

01 Call RACHECK pre-processing exit

Return to RACHECK SVC with code 0 if retry with default profile

A. Switch off flag in ACEEIEP indicating disk data set profile not found

B. Enforce extra restrictions for users with non-standard requirements

C. Set prefix to be checked equal to first 3 characters of disk data set name and transfer to step 1E if not equal to userid

D. Terminate RACHECK successfully by returning with code 8 unless required access is ALTER or CSA option was specified

E. Terminate RACHECK successfully by returning with code 8 if READ access to commonly used system data sets was requested

F. Prompt operator for authority to access sensitive system data sets

G. Change data set name for a disk GDG data set to the name of the GDG base. Set the no profile found bit in the ACEEIEP

Return from pre-processing exit with code 0

02 Check the authorization of the user in the profile of the disk data set or of the tape volume

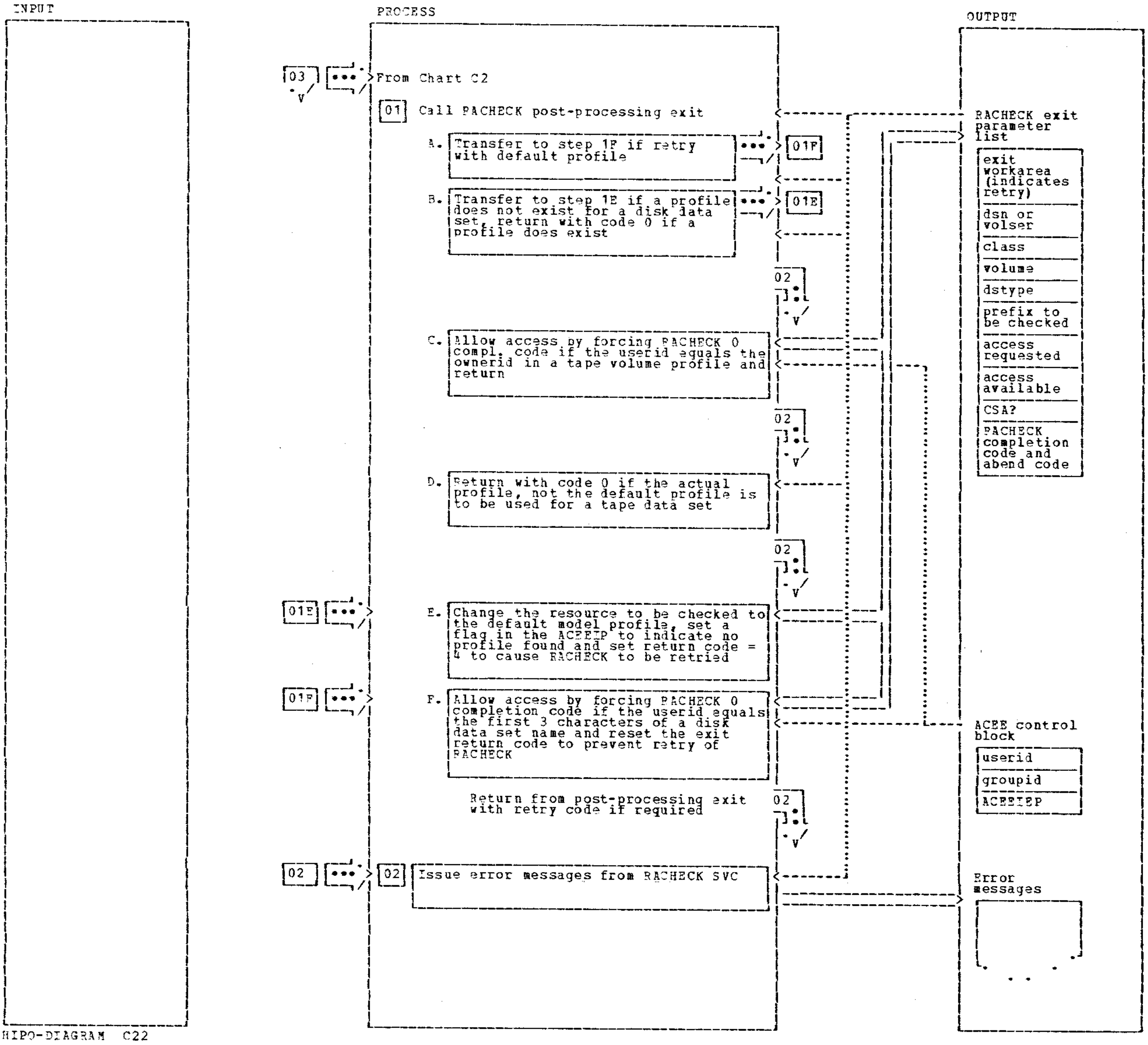
HIPO-DIAGRAM C21

Detailed actions of the RACHECK pre-processing exit for for data set access

	NOTES	MODULE	LABEL	REF
1	Return code 0 allows the RACHECK to proceed normally	ICHRCX01		
1A	The flag is used by the RACDEF pre-processing exit to avoid attempts by RACDEF to delete or alter profiles for disk data sets which do not have profiles.			
1B	Users attached to certain groups are not permitted to access data sets other than their own and system data sets.			
1C	A user in this installation may own data sets with a prefix longer than his 3 character userid as long as the first 3 characters of the prefix equal the userid.			
1D	This is a fast path for RACHECK for a user's own data set. However the full RACHECK must be performed for the CSA option since a copy of the profile is required in storage. Since the no profile flag must be set for data set delete or rename the full RACHECK must be performed for ALTER access (needed for delete or rename). To avoid fast path within the RACHECK SVC for a user's own data set, the prefix to be checked is changed to blank.			
1E	This is a fast-path for RACHECK.			
1F	Authority is required for greater than READ access for most system data sets and for READ access to several. To avoid multiple operator replies in the same job for the same data set, the data set names are chained in storage areas connected to the password area pointed to by the ACEEIEP (The password area is created by the RACINIT exit). The list of data set names is searched every time to avoid an operator reply if possible.			
1G	The volume is changed to dummy. Disk GDG data sets have access controlled by a profile defined for the GDG base or, if this is not defined, by the default profile. The no profile found flag must be set on since the GDG base profile must not be deleted if a generation is deleted.			

HIPO-DIAGRAM C21

Detailed actions of the RACHECK post-processing exit for data set access



01 Call RACHECK post-processing exit

A. Transfer to step 1F if retry with default profile

B. Transfer to step 1E if a profile does not exist for a disk data set, return with code 0 if a profile does exist

C. Allow access by forcing RACHECK 0 compl. code if the userid equals the ownerid in a tape volume profile and return

D. Return with code 0 if the actual profile, not the default profile is to be used for a tape data set

E. Change the resource to be checked to the default model profile, set a flag in the ACEEIP to indicate no profile found and set return code = 4 to cause RACHECK to be retried

F. Allow access by forcing RACHECK 0 completion code if the userid equals the first 3 characters of a disk data set name and reset the exit return code to prevent retry of RACHECK

Return from post-processing exit with retry code if required

02 Issue error messages from RACHECK SVC

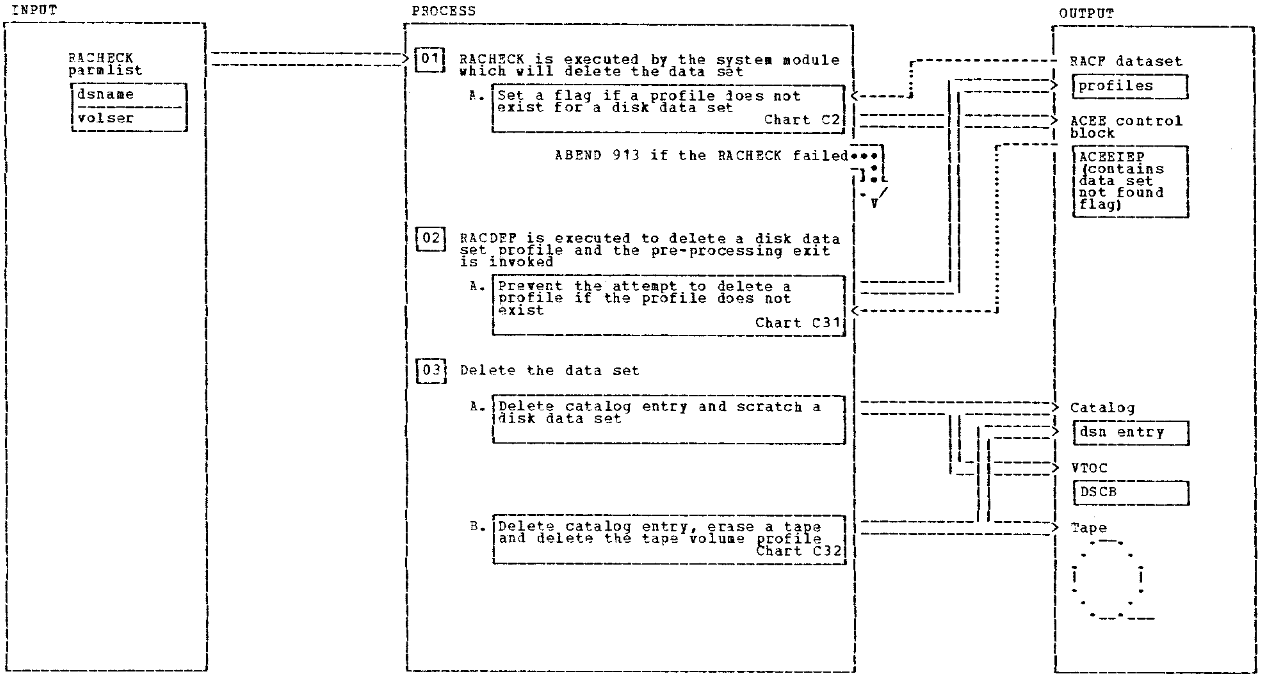
HIPO-DIAGRAM C22

Detailed actions of the RACHECK post-processing exit for data set access

	NOTES	MODULE	LABEL	REF
1A	A return code of 0 is supplied for the subsequent return from the exit to prevent any further attempts at retry by the RACHECK exits i.e. loops are prevented.	ICHRCX02		
1B	Return code 0 allows normal RACHECK to continue. Most disk data sets do not have profiles but are controlled by a default profile for each user.			
1C	The userid is stored in the installation data of a tape profile by the RACDEF exit when the profile is created. Profiles exist for all OLD tape data sets since the exit issues a RACDEF to create a tape profile if one does not exist for any tape data set - this happens when RACHECK occurs during creation of the tape data set. The return code and abend code which would be issued by the RACHECK SVC are altered to 0.			
1D	The installation data contains a flag which indicates whether access to the tape is controlled by the default profile of the tape owner or by the actual tape volume profile.			
1E	The volume is changed to DUMMY, the class to DATASET, the data set type to non-VSAM.			
1F	This step allows access if the prefix was set to blank in step 1D, Chart C21 (see note) and prevents retry with the model profile in this case for a user's own data set.			
2	No error messages are issued by the RACHECK SVC when the disk data set profile is not found, because the retry finds the default profile before entering this step.			

HIPO-DIAGRAM C22

Delete a data set


INPUT
PROCESS
OUTPUT
RACHECK parmlist
dsname
volser
01 RACHECK is executed by the system module which will delete the data set
A. Set a flag if a profile does not exist for a disk data set Chart C2
ABEND 913 if the RACHECK failed
02 RACDEF is executed to delete a disk data set profile and the pre-processing exit is invoked
A. Prevent the attempt to delete a profile if the profile does not exist Chart C31
03 Delete the data set
A. Delete catalog entry and scratch a disk data set
B. Delete catalog entry, erase a tape and delete the tape volume profile Chart C32
RACF dataset
profiles
ACEE control block
ACEEIEP (contains data set not found flag)
Catalog
dsn entry
VTOC
DSCB
Tape

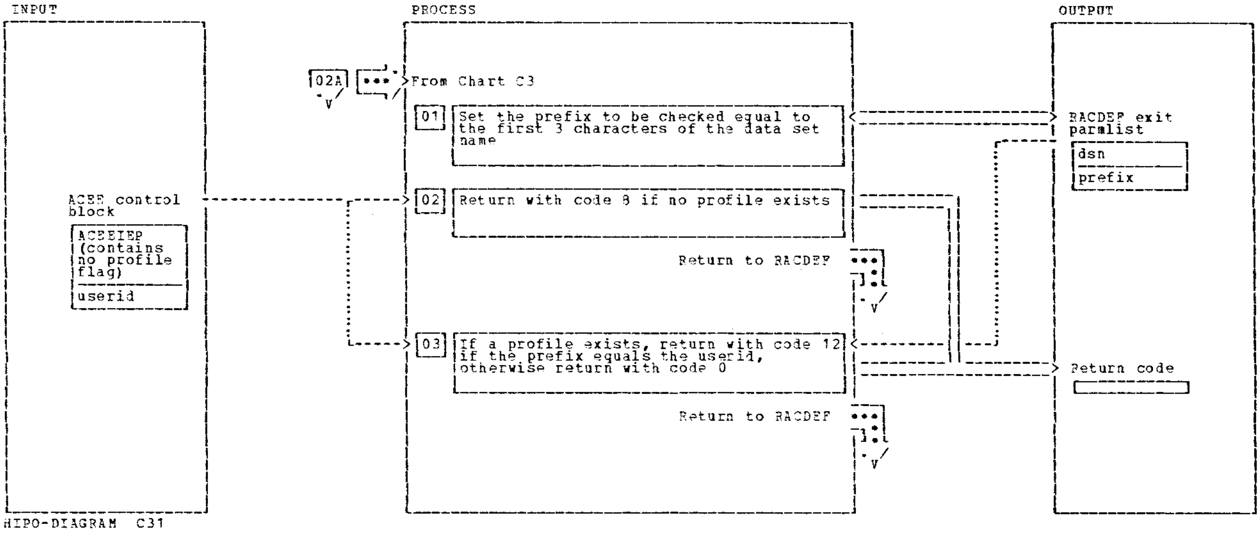
HIPO-DIAGRAM C3

Delete a data set

NOTES	MODULE	LABEL	REF
1A In this installation most disk data sets do not have profiles and access to these data sets is controlled by a default profile defined for each user.			
2A The attempt to delete a non-existent profile would cause a failure of the delete program.			
3B The tape erase and volume profile delete are carried out later by a house-keeping program.			

HIPO-DIAGRAM C3

Prevent attempt to delete non-existent data set profile


INPUT
PROCESS
OUTPUT
02A
From Chart C3
01 Set the prefix to be checked equal to the first 3 characters of the data set name
ACEE control block
ACEEIEP (contains no profile flag)
userid
02 Return with code 8 if no profile exists
Return to RACDEF
03 If a profile exists, return with code 12 if the prefix equals the userid, otherwise return with code 0
Return to RACDEF
RACDEF exit parmlist
dsn
prefix
Return code

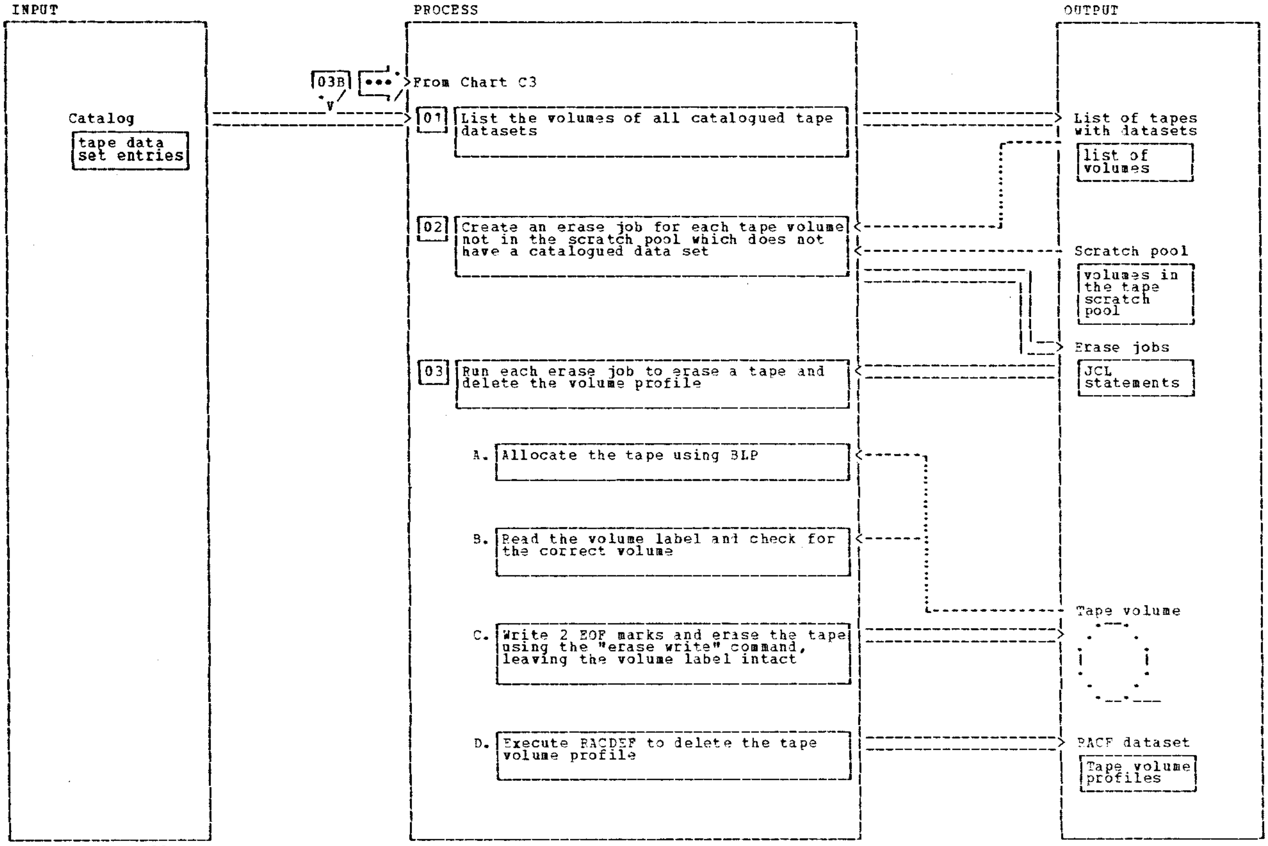
HIPO-DIAGRAM C31

Prevent attempt to delete non-existent data set profile

NOTES	MODULE	LABEL	REF
1 In this installation, a user may own datasets with a longer prefix than his 3 character userid as long as the first 3 characters equal the userid.	ICHRDX01		
2 The return code 8 causes the RACDEF to terminate without attempting to delete the profile. The RACDEF issues a zero completion code to its caller.			
3 Return code 12 causes authorization checking in the RACDEF to be bypassed. Thus users who own datasets with longer prefixes than 3 characters are able to delete them.			

HIPO-DIAGRAM C31

Delete a tape volume profile and erase the tape


INPUT
PROCESS
OUTPUT
03B
From Chart C3
Catalog
tape data set entries
01 List the volumes of all catalogued tape datasets
List of tapes with datasets
list of volumes
02 Create an erase job for each tape volume not in the scratch pool which does not have a catalogued data set
Scratch pool
volumes in the tape scratch pool
Erase jobs
JCL statements
03 Run each erase job to erase a tape and delete the volume profile
A. Allocate the tape using BLP
B. Read the volume label and check for the correct volume
Tape volume
C. Write 2 EOF marks and erase the tape using the "erase write" command, leaving the volume label intact
D. Execute RACDEF to delete the tape volume profile
RACF dataset
Tape volume profiles

HIPO-DIAGRAM C32

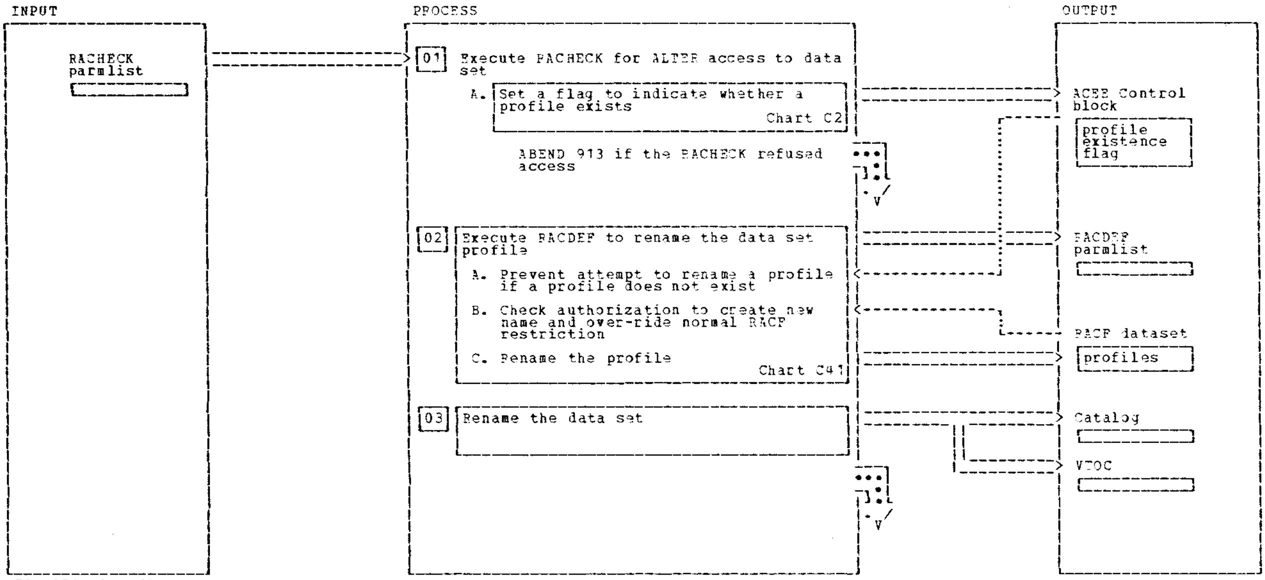
Delete a tape volume profile and erase the tape

	NOTES	MODULE	LABEL	REF
1	All tape data sets stored on the standard range of volumes are catalogued.			
2	A list of tapes in the scratch pool - i.e. with no data stored on them is maintained. The tapes considered are a standard range of tapes stored near the computer room which can be used for scratch or to store permanent data sets. The list of volumes with catalogued data sets is compared with the list of volumes not in the scratch pool.			

	NOTES	MODULE	LABEL	REF
3A	BLP allows the label to be processed as a data file.			
3B	Since normal label checking is bypassed by BLP, the program checks the label.			
3C	"Erase write" only involves the tape drive, not the control unit or channel.			
3D	The RACDEF exit allows the RACDEF to proceed (the erase program must be authorized to be able to execute RACDEF).			

HIPO-DIAGRAM C32

Rename a disk data set


INPUT
PROCESS
OUTPUT
RACHECK parmlist
01 Execute RACHECK for ALTER access to data set
A. Set a flag to indicate whether a profile exists
Chart C2
ACEE Control block
profile existence flag
ABEND 913 if the RACHECK refused access
02 Execute RACDEF to rename the data set profile
A. Prevent attempt to rename a profile if a profile does not exist
B. Check authorization to create new name and over-ride normal RACF restriction
C. Rename the profile
Chart C41
RACDEF parmlist
RACF dataset
profiles
03 Rename the data set
Catalog
VTOC

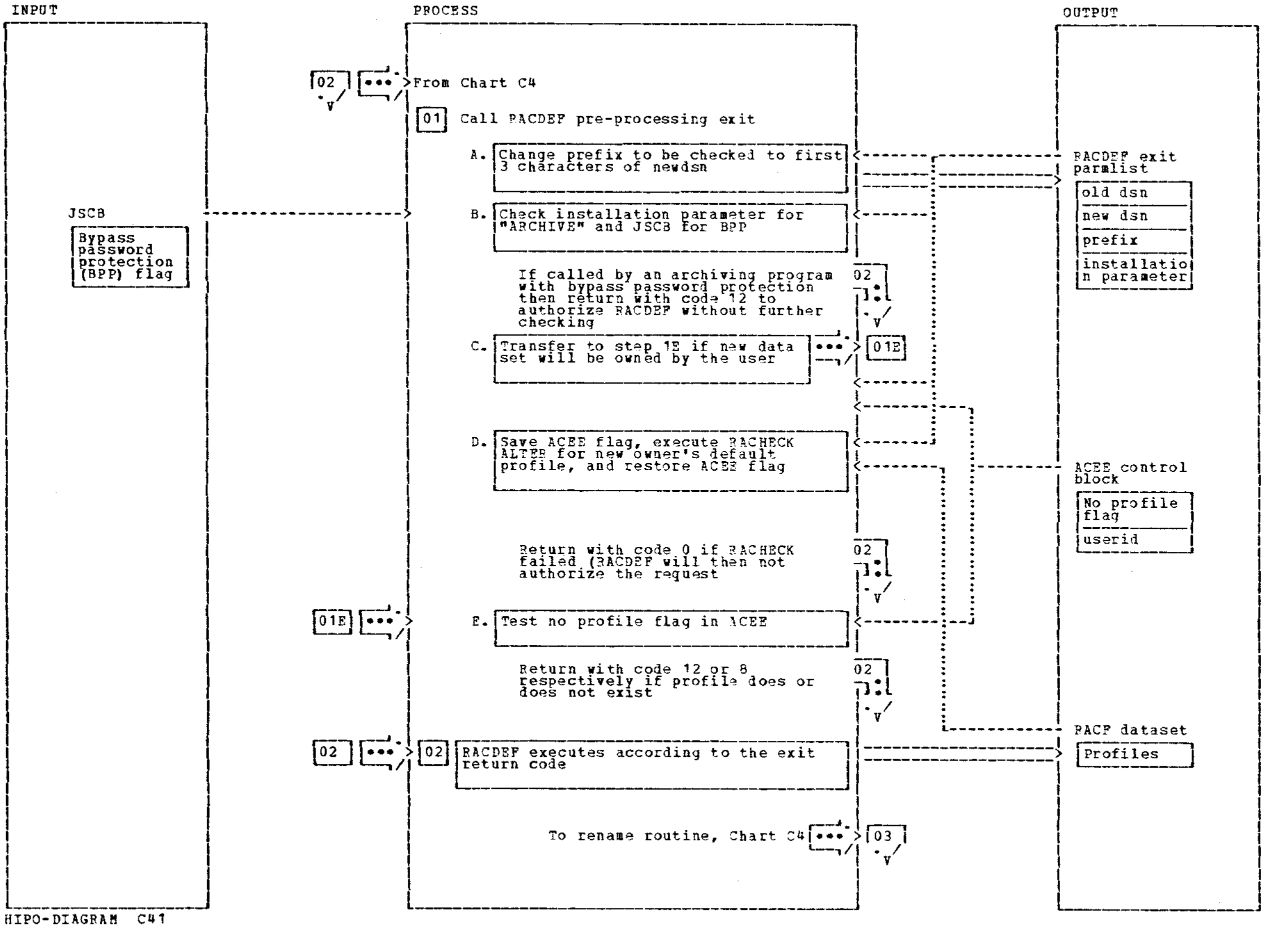
HIPO-DIAGRAM C4

Rename a disk data set

	NOTES	MODULE	LABEL	REF
1A	Most data sets in this installation do not have profiles but access to them is controlled by a default profile for each user.			
2B	RACF normally does not allow a user to create a data set for another user. The existence of a default profile for each user in this installation allows this rule to be relaxed so that users may create data sets for other users if they are given ALTER access authority in the other users default profile.			
2A	The attempt to rename a non-existent profile would cause the entire rename to fail if it was allowed to be attempted.			

HIPO-DIAGRAM C4

RACDEF rename for a disk data set profile



01 Call RACDEF pre-processing exit

A. Change prefix to be checked to first 3 characters of newdsn

B. Check installation parameter for "ARCHIVE" and JSCB for BPP

If called by an archiving program with bypass password protection then return with code 12 to authorize RACDEF without further checking

C. Transfer to step 1E if new data set will be owned by the user

D. Save ACEE flag, execute RACHECK ALTER for new owner's default profile, and restore ACEE flag

Return with code 0 if RACHECK failed (RACDEF will then not authorize the request

E. Test no profile flag in ACEE

Return with code 12 or 8 respectively if profile does or does not exist

02 RACDEF executes according to the exit return code

To rename routine, Chart C4

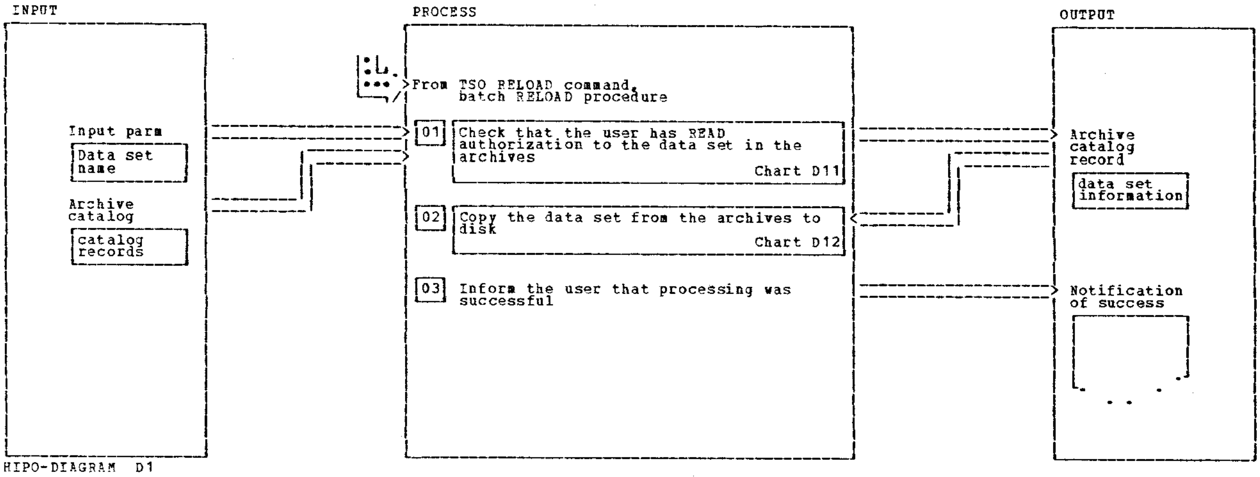
HIPO-DIAGRAM C41

RACDEF rename for a disk data set profile

	NOTES	MODULE	LABEL	REF
1A	Users in this installation may own data sets prefixed by longer than 3 characters as long as the first 3 characters equal their 3 character userid.	ICHRDX01		
1B	Archive programs are authorized and use RACHECK to determine if a profile exists before executing RACDEF.			
1C	The RACDEF may be allowed to proceed if the user will own the new data set since an RACHECK has already determined that he has ALTER access to the old data set.			C4
1D	The ACEE flag has to be saved and restored because the RACHECK will destroy it. RACHECK ALTER for the new owners default profile is appropriate since no specific definition of the data set by the new owner can exist at this stage.			
1E	The no profile flag in the ACEE is set by a RACHECK exit in the RACHECK executed prior to the execution of the RACDEF. It indicates whether the data set has a profile. Data sets without profiles in this installation have access controlled by a default profile for each user.			C4
2	Return code 8 from the exit prevents any further action by RACDEF but causes RACDEF to appear to complete successfully. It is used to avoid problems when a profile is not defined for a data set. Return code 12 from the exit causes the RACDEF to continue normally except that it's normal authorization checking is bypassed.			

HIPO-DIAGRAM C41

Reload a Data Set from the Archives



INPUT

Input parm

Data set name

Archive catalog

catalog records

PROCESS

From TSO RELOAD command, batch RELOAD procedure

01 Check that the user has READ authorization to the data set in the archives Chart D11

02 Copy the data set from the archives to disk Chart D12

03 Inform the user that processing was successful

OUTPUT

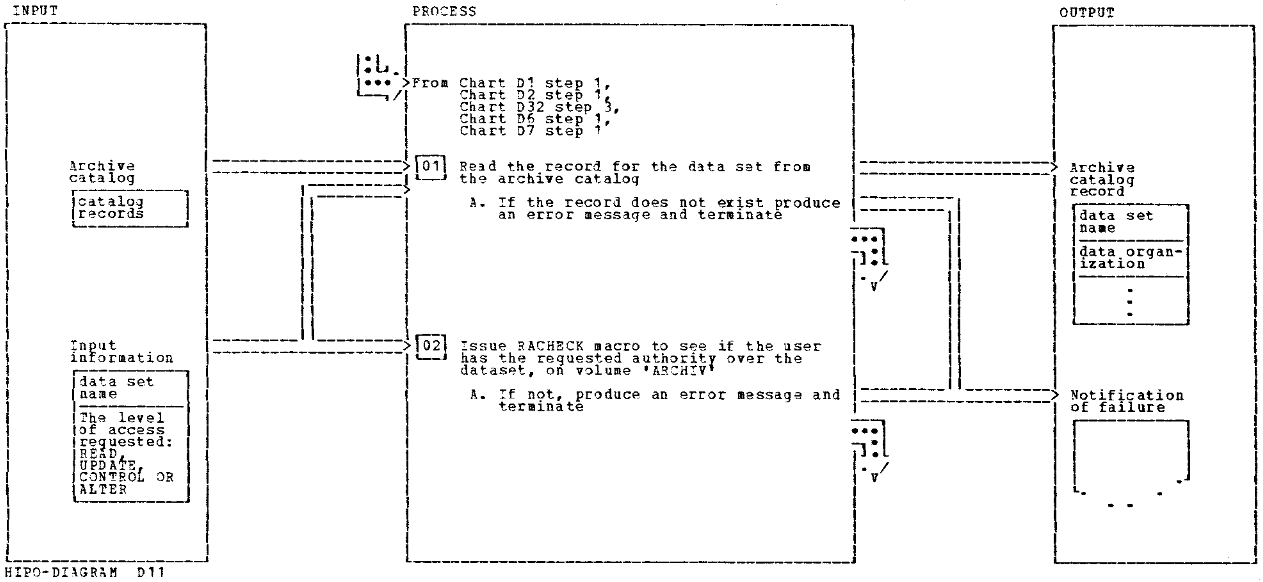
Archive catalog record

data set information

Notification of success

HIPO-DIAGRAM D1

Check Authorization to a Data Set in the Archives



INPUT

Archive catalog

catalog records

Input information

data set name

The level of access requested: READ, UPDATE, CONTROL OR ALTER

PROCESS

From Chart D1 step 1, Chart D2 step 1, Chart D32 step 3, Chart D6 step 1, Chart D7 step 1

01 Read the record for the data set from the archive catalog

A. If the record does not exist produce an error message and terminate

02 Issue RACHECK macro to see if the user has the requested authority over the dataset, on volume 'ARCHIV'

A. If not, produce an error message and terminate

OUTPUT

Archive catalog record

data set name

data organ- ization

⋮

Notification of failure

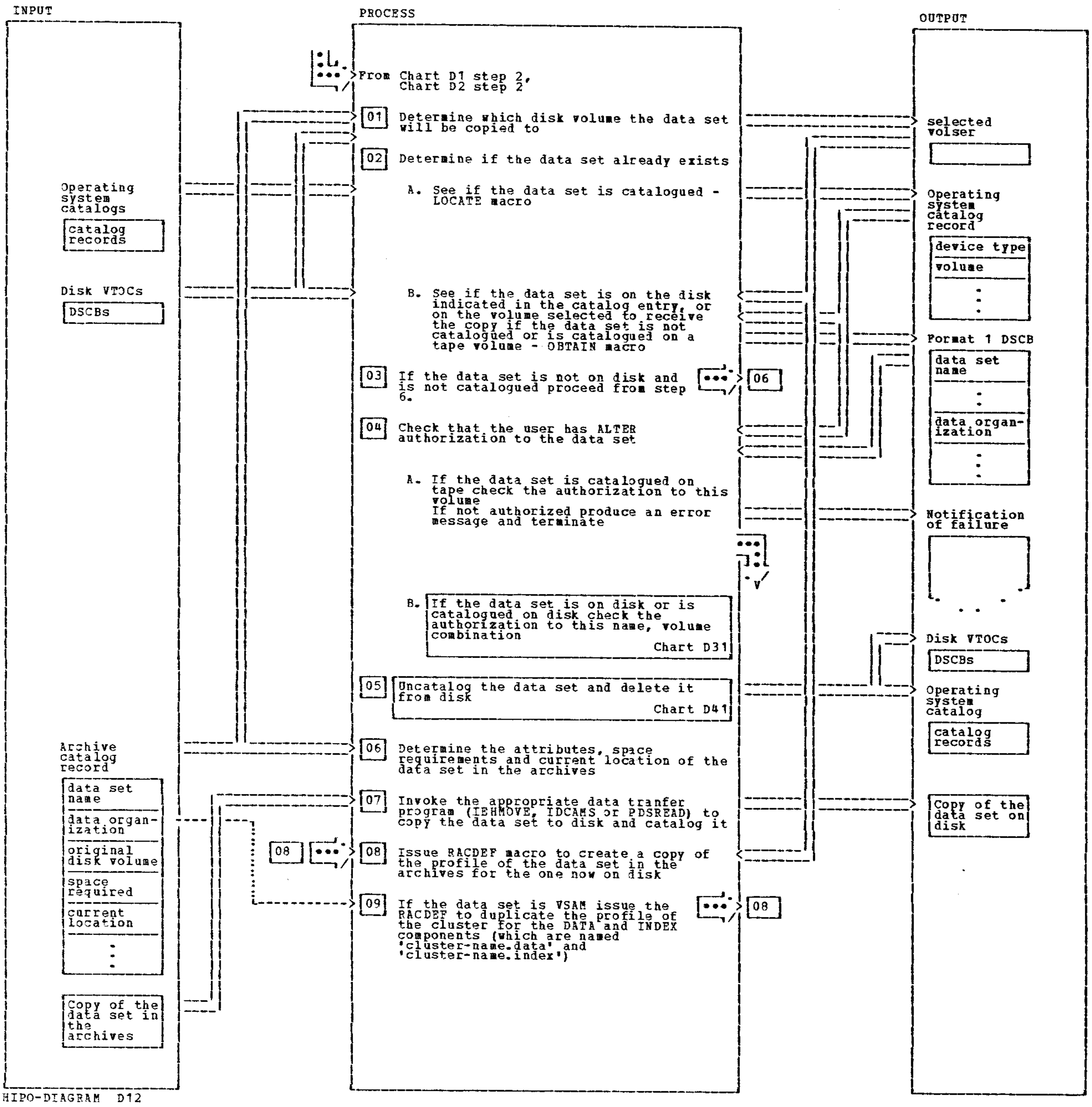
HIPO-DIAGRAM D11

Check Authorization to a Data Set in the Archives

NOTES	MODULE	LABEL	REF	NOTES	MODULE	LABEL	REF
2 Specific profiles for all data sets in the archives have 'ARCHIV' in the volume field. This imaginary volume just serves to distinguish between data sets of the same name in archives and on disk.							

HIPO-DIAGRAM D11

Copy a Data Set from the Archives to Disk



INPUT

Operating system catalogs

catalog records

Disk VTOCs

DSCBs

Archive catalog record

data set name

data organ-ization

original disk volume

space required

current location

:

Copy of the data set in the archives

PROCESS

From Chart D1 step 2, Chart D2 step 2

01 Determine which disk volume the data set will be copied to

02 Determine if the data set already exists

A. See if the data set is catalogued - LOCATE macro

B. See if the data set is on the disk indicated in the catalog entry, or on the volume selected to receive the copy if the data set is not catalogued or is catalogued on a tape volume - OBTAIN macro

03 If the data set is not on disk and is not catalogued proceed from step 6. → 06

04 Check that the user has ALTER authorization to the data set

A. If the data set is catalogued on tape check the authorization to this volume
If not authorized produce an error message and terminate

B. If the data set is on disk or is catalogued on disk check the authorization to this name, volume combination
Chart D31

05 Uncatalog the data set and delete it from disk
Chart D41

06 Determine the attributes, space requirements and current location of the data set in the archives

07 Invoke the appropriate data tranfer program (IEHMOVE, IDCAMS or PDSREAD) to copy the data set to disk and catalog it

08 → 08 Issue RACDEF macro to create a copy of the profile of the data set in the archives for the one now on disk

09 If the data set is VSAM issue the RACDEF to duplicate the profile of the cluster for the DATA and INDEX components (which are named 'cluster-name.data' and 'cluster-name.index') → 08

OUTPUT

selected volser

Operating system catalog record

device type

volume

:

Format 1 DSCB

data set name

:

data organ-ization

:

Notification of failure

Disk VTOCs

DSCBs

Operating system catalog

catalog records

Copy of the data set on disk

HIPO-DIAGRAM D12

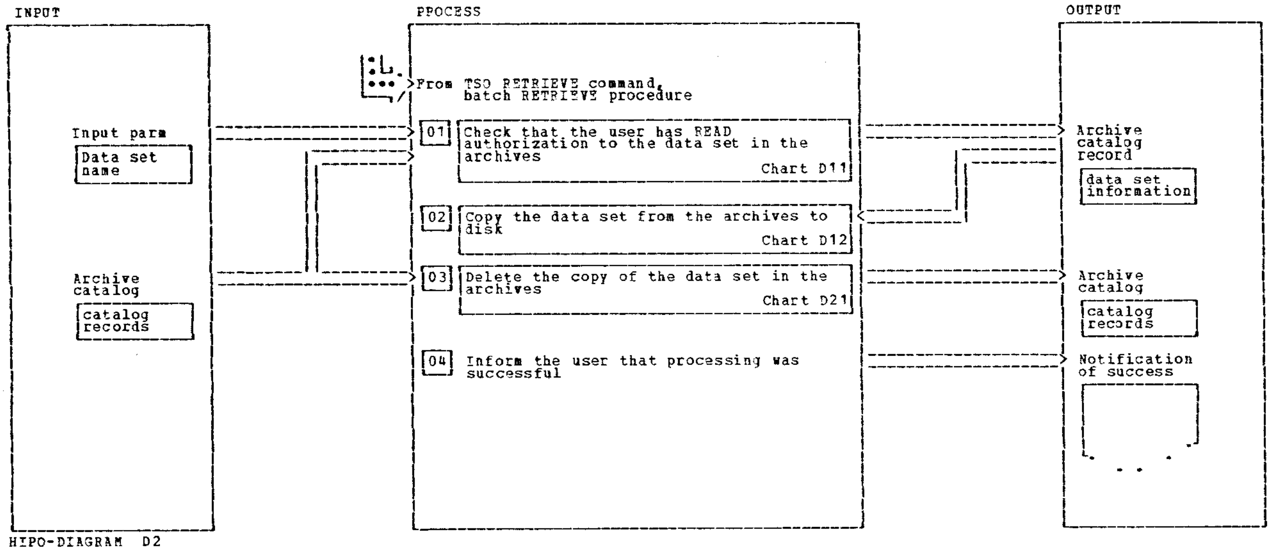
Copy a Data Set from the Archives to Disk

	NOTES	MODULE	LABEL	REF
1	The data set will be returned to the disk volume it came from if it occupies 1 cylinder or less. Otherwise the volume with the largest amount of free space will be selected.			
2	An existing data set of the same name may have to be deleted first.			
4	If an existing data set is being deleted or uncatalogued the user must have ALTER authority to this version.			
6	The main attribute is the data set type - sequential, partitioned, direct access or VSAM.			
6	Some archived data sets reside on tape and some in a special partitioned data set on disk. Different programs are required for the various data set type/storage medium combinations.			

	NOTES	MODULE	LABEL	REF
7	The RACDEF attempts to model the profile of the archived data set. If the archived data set doesn't have a specific profile (it is protected by the user's default profile) then the RACDEF will fail and will not create a profile for the disk data set, causing it to be protected by the user's default as well.			
8	VSAM data sets may have DATA and INDEX components which have the same protection requirements as the cluster. Their names are governed by an installation standard.			

HIPO-DIAGRAM D12

Retrieve a Data Set from the Archives



INPUT

Input parm

Data set name

Archive catalog

catalog records

PROCESS

From TSO RETRIEVE command, batch RETRIEVE procedure

01 Check that the user has READ authorization to the data set in the archives Chart D11

02 Copy the data set from the archives to disk Chart D12

03 Delete the copy of the data set in the archives Chart D21

04 Inform the user that processing was successful

OUTPUT

Archive catalog record

data set information

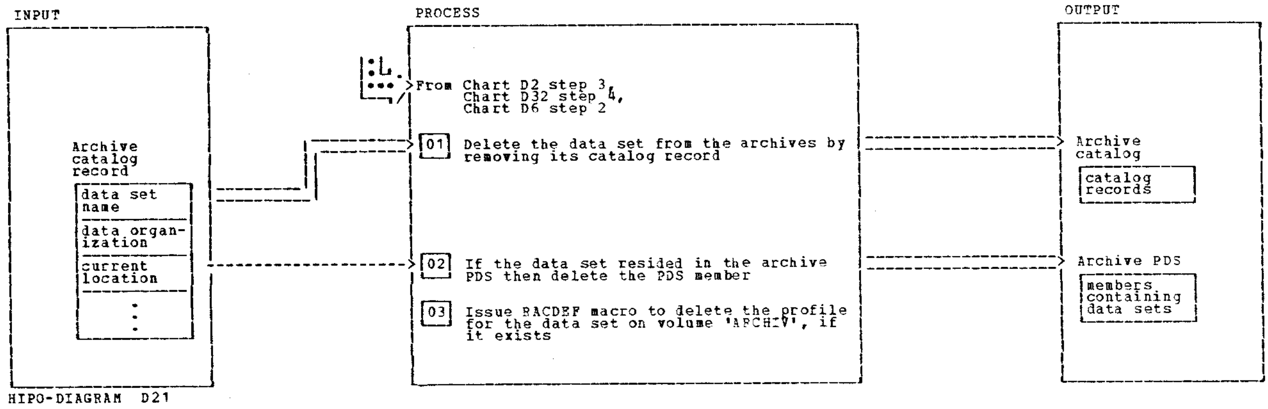
Archive catalog

catalog records

Notification of success

HIPO-DIAGRAM D2

Delete a Data Set from the Archives



INPUT

Archive catalog record

data set name

data organization

current location

⋮

PROCESS

From Chart D2 step 3, Chart D32 step 4, Chart D6 step 2

01 Delete the data set from the archives by removing its catalog record

02 If the data set resided in the archive PDS then delete the PDS member

03 Issue RACDEF macro to delete the profile for the data set on volume 'ARCHIV', if it exists

OUTPUT

Archive catalog

catalog records

Archive PDS

members containing data sets

HIPO-DIAGRAM D21

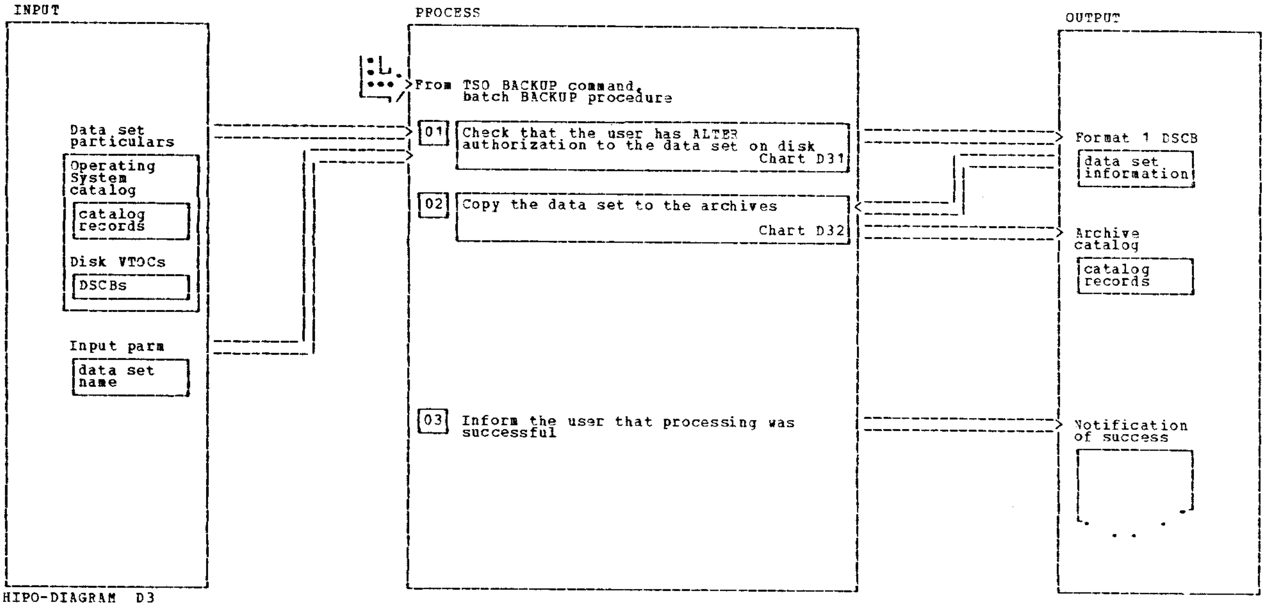
Delete a Data Set from the Archives

NOTES	MODULE	LABEL	REF
1 A data set is deleted from the archives by simply removing reference to it from the Archive catalog. 2 If the data set is in the special archive PDS the associated member is also deleted, primarily to enable the disk space to be reclaimed.			

NOTES	MODULE	LABEL	REF
3 If entered from Chart D2 then the RACDEF issued at step 8 of Chart D12 will have already indicated whether a specific profile exists or not and an associated return code is available for testing. This RACDEF is bypassed if the return code is non-zero. If entered from Chart D32 then the RACHECK issued at step 3 of Chart D31 will have set the appropriate value in the flag in ACEEIEP indicating whether the profile exists or not. In this case this RACDEF is always issued and the pre-processing exit will bypass SVC processing if the flag is set. A similar situation exists if entered from Chart 6.			

HIPO-DIAGRAM D21

Backup a Data Set to the Archives



INPUT

Data set particulars

Operating System catalog

catalog records

Disk VTOCs

DSCBs

Input parm

data set name

PROCESS

From TSO BACKUP command, batch BACKUP procedure

01 Check that the user has ALTER authorization to the data set on disk
Chart D31

02 Copy the data set to the archives
Chart D32

03 Inform the user that processing was successful

OUTPUT

Format 1 DSCB

data set information

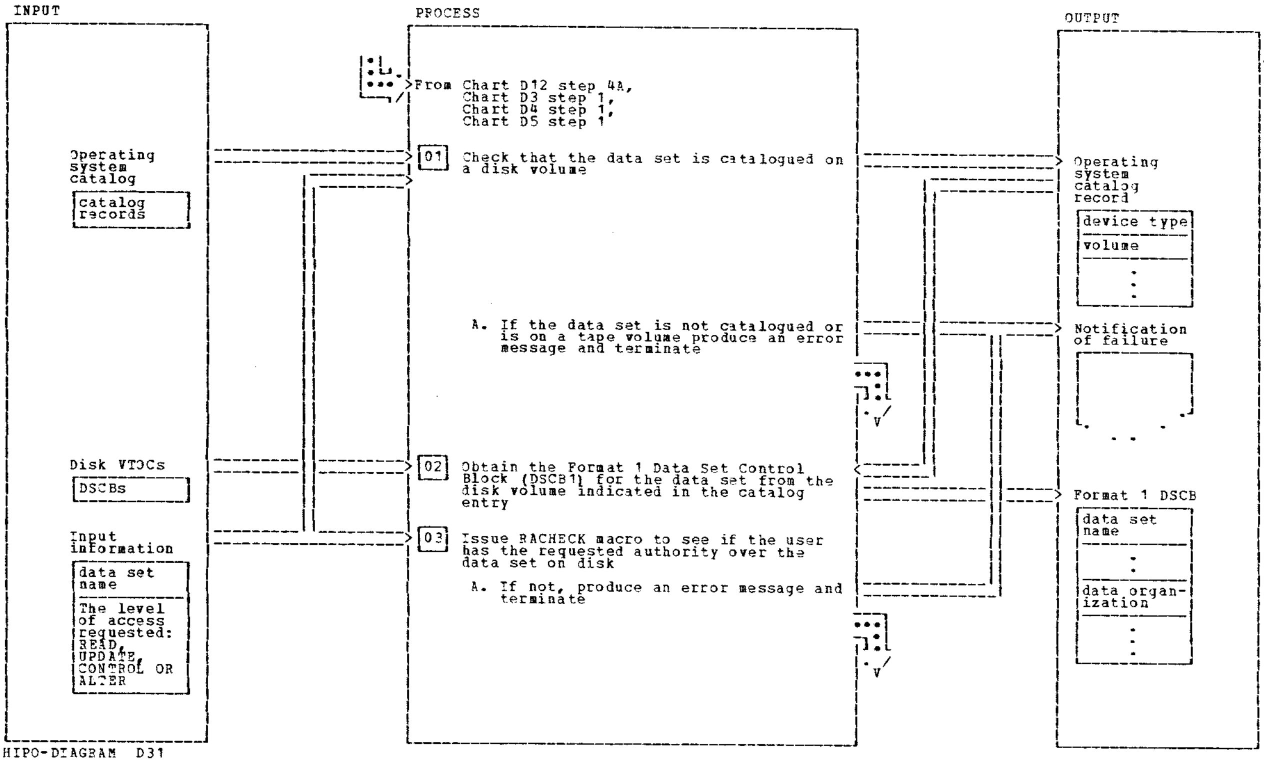
Archive catalog

catalog records

Notification of success

HIPO-DIAGRAM D3

Check Authorization to a Data Set on Disk



INPUT

Operating system catalog

catalog records

Disk VTOCs

DSCBs

Input information

data set name

The level of access requested: READ, UPDATE, CONTROL OR ALTER

PROCESS

From Chart D12 step 4A, Chart D3 step 1, Chart D4 step 1, Chart D5 step 1

01 Check that the data set is catalogued on a disk volume

A. If the data set is not catalogued or is on a tape volume produce an error message and terminate

02 Obtain the Format 1 Data Set Control Block (DSCB1) for the data set from the disk volume indicated in the catalog entry

03 Issue RACHECK macro to see if the user has the requested authority over the data set on disk

A. If not, produce an error message and terminate

OUTPUT

Operating system catalog record

device type

volume

Notification of failure

Format 1 DSCB

data set name

data organization

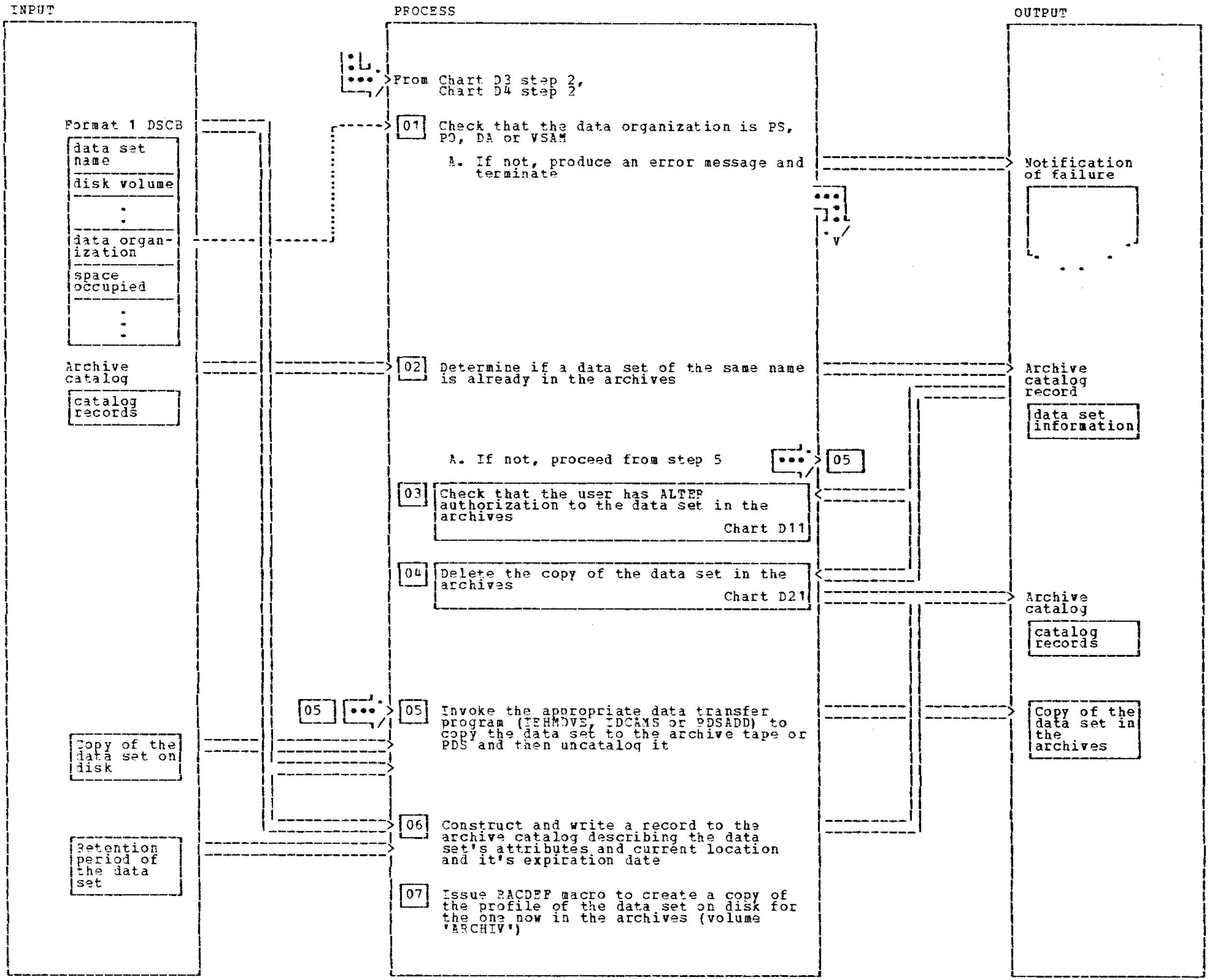
HIPO-DIAGRAM D31

Check Authorization to a Data Set on Disk

NOTES	MODULE	LABEL	REF	NOTES	MODULE	LABEL	REF
1 Only catalogued, disk data sets can be archived. 2 For a VSAM data set the DSCB1 will be incomplete, but will at least indicate that the data set is VSAM.				3 For a VSAM data set the volume containing the catalog entry must be determined and used in the RACHECK, rather than the volume containing the data set.			

HIPO-DIAGRAM D31

Copy a Data Set from Disk to the Archives



01 Check that the data organization is PS, PO, DA or VSAM

A. If not, produce an error message and terminate

02 Determine if a data set of the same name is already in the archives

A. If not, proceed from step 5

03 Check that the user has ALTER authorization to the data set in the archives Chart D11

04 Delete the copy of the data set in the archives Chart D21

05 Invoke the appropriate data transfer program (IEHMOVE, IDCAMS or PDSADD) to copy the data set to the archive tape or PDS and then uncatalog it

06 Construct and write a record to the archive catalog describing the data set's attributes and current location and it's expiration date

07 Issue RACDEF macro to create a copy of the profile of the data set on disk for the one now in the archives (volume 'ARCHIV')

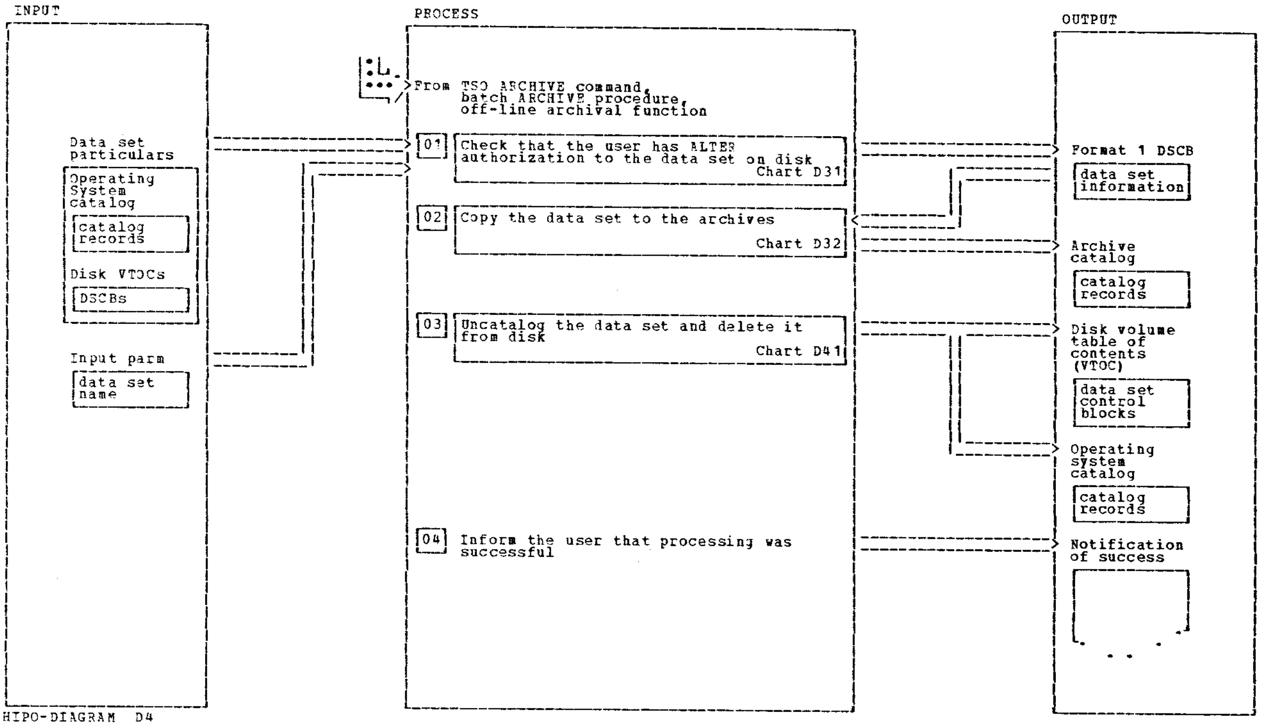
HIPO-DIAGRAM D32

Copy a Data Set from Disk to the Archives

NOTES	MODULE	LABEL	REF	NOTES	MODULE	LABEL	REF
1 These are the only data set types currently supported.				6 The archive catalog record contains all information necessary to return the data set to disk if later required.			
2 If a data set of the same name already exists in the archives it must be deleted first.				7 The RACDEF attempts to model the profile of the disk data set. If the disk data set doesn't have a specific profile (it is protected by the user's default profile) then the RACDEF will fail and will not create a profile for the archived data set, causing it to be protected by the user's default as well.			
3 ALTER authorization is required to delete the copy in the archives.							
5 Some archived data sets reside on tape and some in a special partitioned data set on disk. Different programs are required for the various data set type/storage medium combinations.							

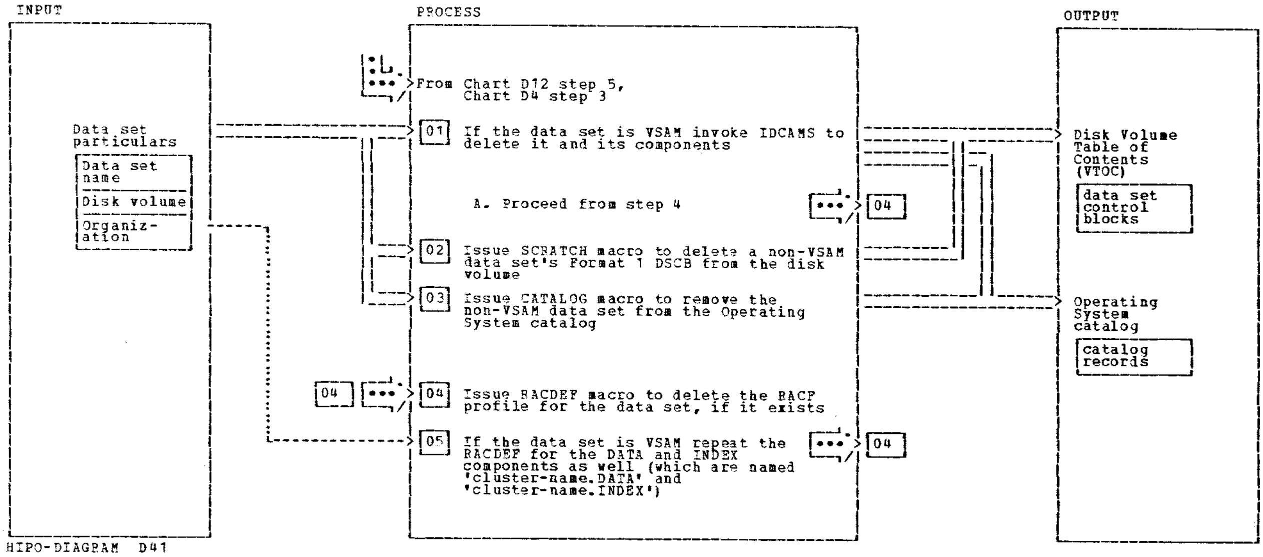
HIPO-DIAGRAM D32

Archive a Data Set


INPUT
PROCESS
OUTPUT
From TSO ARCHIVE command, batch ARCHIVE procedure, off-line archival function
Data set particulars
Operating System catalog
catalog records
Disk VTOCs
DSCBs
Input parm
data set name
01 Check that the user has ALTER authorization to the data set on disk Chart D31
02 Copy the data set to the archives Chart D32
03 Uncatalog the data set and delete it from disk Chart D41
04 Inform the user that processing was successful
Format 1 DSCB
data set information
Archive catalog
catalog records
Disk volume table of contents (VTOC)
data set control blocks
Operating system catalog
catalog records
Notification of success

HIPO-DIAGRAM D4

Delete a Data Set from Disk


INPUT
PROCESS
OUTPUT
From Chart D12 step 5, Chart D4 step 3
Data set particulars
Data set name
Disk volume
Organiz-ation
01 If the data set is VSAM invoke IDCAMS to delete it and its components
A. Proceed from step 4
04
02 Issue SCRATCH macro to delete a non-VSAM data set's Format 1 DSCB from the disk volume
03 Issue CATALOG macro to remove the non-VSAM data set from the Operating System catalog
04
04 Issue RACDEF macro to delete the RACF profile for the data set, if it exists
05 If the data set is VSAM repeat the RACDEF for the DATA and INDEX components as well (which are named 'cluster-name.DATA' and 'cluster-name.INDEX')
04
Disk Volume Table of Contents (VTOC)
data set control blocks
Operating System catalog
catalog records

HIPO-DIAGRAM D41

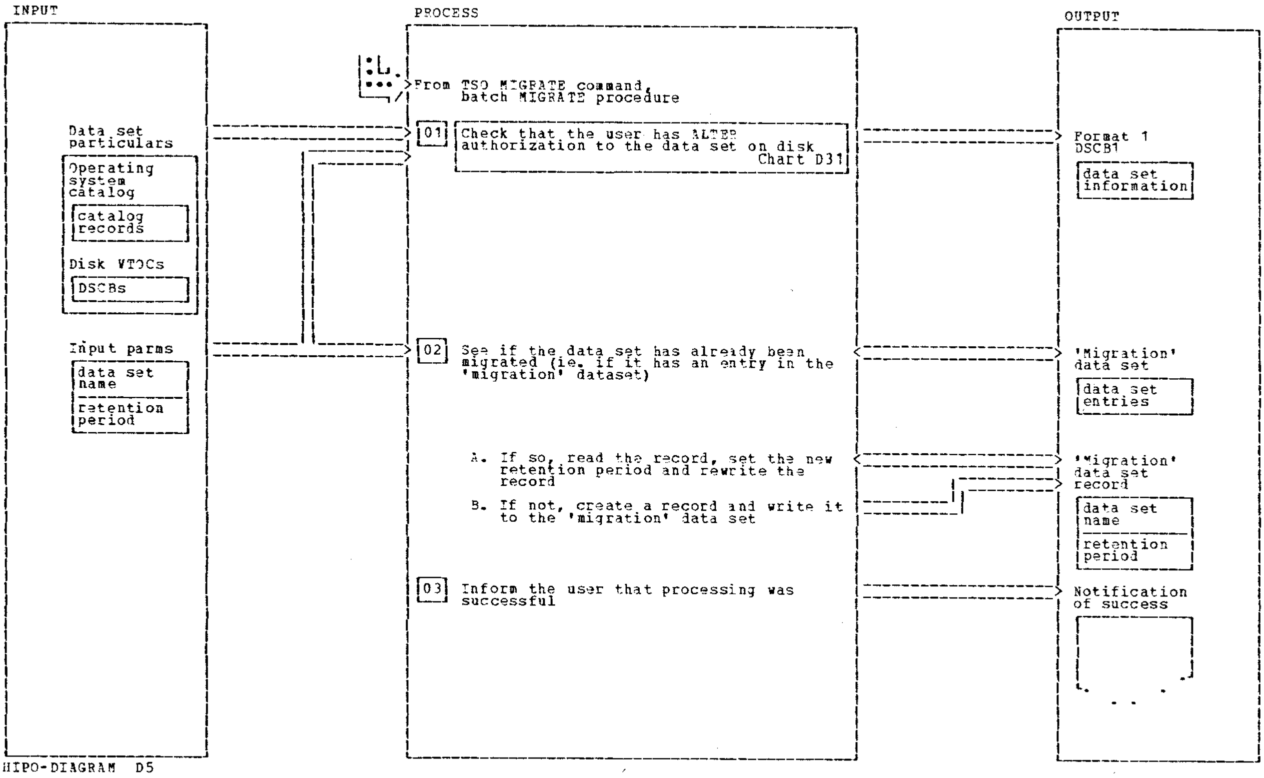
Delete a Data Set from Disk

NOTES	MODULE	LABEL	REF
1 IDCAMS is the IBM utility program that performs a variety of functions for VSAM data sets.			
4 For a VSAM data set the volume containing the catalog entry must be determined and used in the RADEF, rather than the volume containing the data set. If entered from Chart D4 then the RACDEF issued at step 7 of Chart D32 will have already indicated whether a specific profile exists or not and an associated return code is available for testing. This RACDEF is bypassed if the return code is non-zero. If entered from Chart D12 then the RACHECK issued at step 3 of Chart D31 will have set the appropriate value in the flag in ACEEIEP indicating whether the profile exists or not. In this case this RACDEF is always issued and the pre-processing exit will bypass SVC processing if the flag is set.			

NOTES	MODULE	LABEL	REF
5 The DATA and INDEX component names are governed by an installation standard.			

HIPO-DIAGRAM D41

Migrate a Data Set to the Archives


INPUT
PROCESS
OUTPUT
From TSO MIGRATE command, batch MIGRATE procedure
Data set particulars
Operating system catalog
catalog records
Disk VTOCs
DSCBs
Input parms
data set name
retention period
01 Check that the user has ALTER authorization to the data set on disk Chart D31
02 See if the data set has already been migrated (ie. if it has an entry in the 'migration' dataset)
A. If so, read the record, set the new retention period and rewrite the record
B. If not, create a record and write it to the 'migration' data set
03 Inform the user that processing was successful
Format 1 DSCB1
data set information
'Migration' data set
data set entries
'Migration' data set record
data set name
retention period
Notification of success

HIPO-DIAGRAM D5

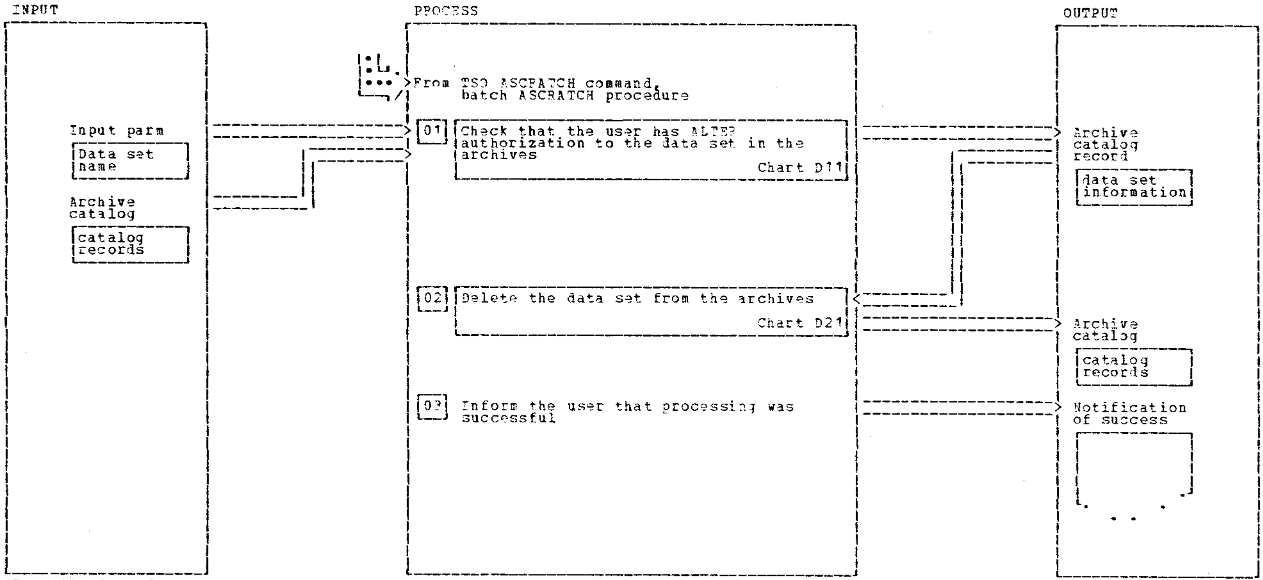
Migrate a Data Set to the Archives

NOTES	MODULE	LABEL	REF
1 Although no data access is involved in the MIGPATE operation ALTER access is implied by its function.			

NOTES	MODULE	LABEL	REF
2 The 'migration' data set contains an entry for each data set that has been migrated (flagged for off-line archival). The entry contains the data set name and an initial retention period for it.			

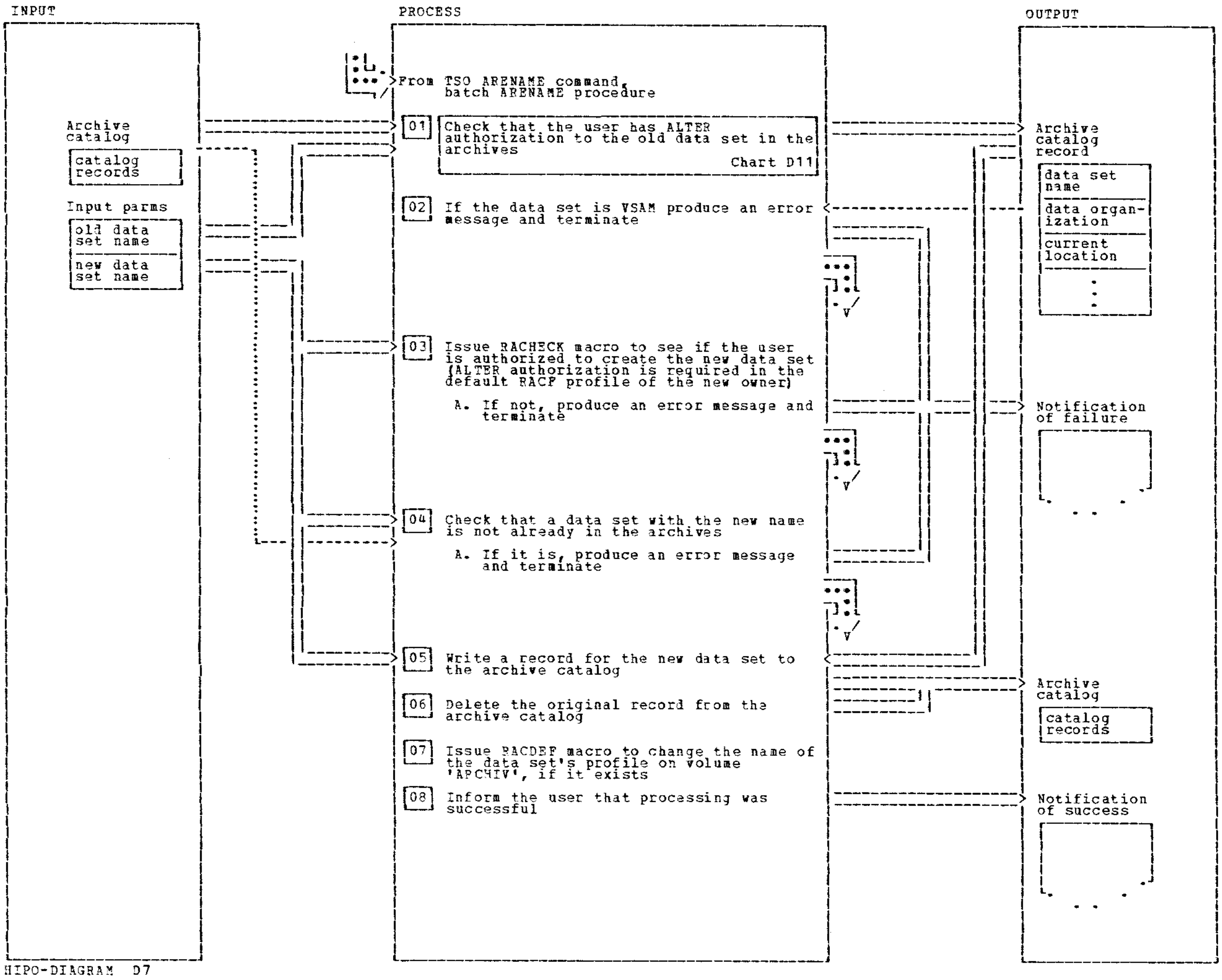
HIPO-DIAGRAM D5

Scratch a Data Set from the Archives


INPUT
PROCESS
OUTPUT
From TSO ASCRATCH command, batch ASCRATCH procedure
Input parm
Data set name
Archive catalog
catalog records
01 Check that the user has ALTER authorization to the data set in the archives Chart D11
02 Delete the data set from the archives Chart D21
03 Inform the user that processing was successful
Archive catalog record
data set information
Archive catalog
catalog records
Notification of success

HIPO-DIAGRAM D6

Rename a Data Set in the Archives



01 Check that the user has ALTER authorization to the old data set in the archives
Chart D11

02 If the data set is VSAM produce an error message and terminate

03 Issue RACHECK macro to see if the user is authorized to create the new data set (ALTER authorization is required in the default RACF profile of the new owner)

A. If not, produce an error message and terminate

04 Check that a data set with the new name is not already in the archives

A. If it is, produce an error message and terminate

05 Write a record for the new data set to the archive catalog

06 Delete the original record from the archive catalog

07 Issue RACDEF macro to change the name of the data set's profile on volume 'ARCHIV', if it exists

08 Inform the user that processing was successful

HIPO-DIAGRAM D7

Rename a Data Set in the Archives

NOTES	MODULE	LABEL	REF
2 VSAM data sets cannot be renamed while in the archives due to VSAM catalog volume ownership implications.			
3 This function is provided for consistency with the ability to create a disk data set for another user - see Charts C1 and C4.			

NOTES	MODULE	LABEL	REF
7 The RACDEF NEWNAME option is used for this function. If the old name doesn't have a specific profile (it is protected by the user's default profile) then the RACDEF will fail and will not create a profile for the new data set, causing it to be protected by the default as well.			

HIPO-DIAGRAM D7

APPENDIX V

LISTINGS OF RACF EXITS AND OTHER PROGRAMS

Definitions of the flags used in the RACF exits

Control block	Displacement	Size	Bit	Exit	
ACEE	+12(ACEEIEP)	1	x0000000	ICHRCX02 ICHRDX01	indicates no profile exists for a disk data set
ACEE	+13(ACEEIEP)	3		ICHRIX01 ICHRCX01 ICHRIX02	points to an area containing the password and pointing to areas containing data set names
exit work area	+0	1	x0000000	ICHRIX01 ICHRIX02	indicates that password should not be checked
exit work area	+1	1	x0000000	ICHRIX01 ICHRIX02	indicates retry in progress
exit work area	+2	1	x0000000	ICHRIX01 ICHRIX02	indicates that RACINIT should be failed
exit work area	+0	1	x0000000	ICHRCX01 ICHRCX02	access allowed by pre-processing exit
exit work area	+1	1	x0000000	ICHRCX01 ICHRCX02	indicates retry of RACHECK with default profile

Definitions of installation parameters used in exits

SVC	Parameter content	Use
RACDEF	'ARCHIVE'	indicates that SVC was issued by an archive program
RACDEF	dsn	RACHECK post-processing exit has issued RACDEF to create a profile for a new tape data set
RACHECK	dsn	OPEN has issued a RACHECK during the creation of a new tape data set

```
*
*                    RACDEF PRE-PROCESSING EXIT
*
ICHRDX01 START 0
         SAVE  (14,12),,*
         LR    12,15
         USING ICHRDX01,12
         LR    2,1            RACDEF EXIT PARM LIST ADDR
         L     4,16           CVT
         L     4,0(4)         CVTTCBP
         L     4,12(4)        ASCB
         L     4,108(4)       ASXB
         L     10,200(4)      ACEE
         XR    15,15       RC IF NO ACEE
         LTR   10,10
         BZ    RETURNB      NO ACEE - NOT RACF DEFINED USER
         L     5,12(10)           ACEEIEP
         LA    5,0(5)
         LTR   5,5
         BZ    GETCLASS
         MVI   77(5),X'00'    INDICATE NO LONGER RACFDEF RENAME
GETCLASS L     3,24(2)            CLASS
         CLC   =C'DATASET',1(3)
         BNE   TEST
         L     3,12(2)               DSN
         L     4,4(2)                FLAG
         TM    0(4),X'10'            NEWNAME ?
         BZ    GETCMND               NO
         L     3,16(2)               NEWNAME ADDRESS
GETCMND  L     4,40(2)               CMMND PARMS
         L     4,32(4)               PREFIX
         MVC   0(3,4),0(3)
         MVC   3(5,4),=CL5' '   SET PREFIX = 1ST 3 CHARS OF DSN
TEST     L     3,4(2)
         LTR   3,3
         BZ    ABEND1
         TM    0(3),X'C0'
         BM    DELETE           DELETE OR ADDVOL
*
*           RACDEF DEFINE
*
DEFINE   L     3,24(2)        RESOURCE CLASS ADDR
         LTR   3,3
         BZ    ABEND2
         CLC   =C'TAPEVOL',1(3)
         BE    RACHTAPE
         CLC   =C'DATASET',1(3)
         BNE   CONTINUE       OTHER THAN TAPE OR DISK
*
*           DEFINE OR RENAME DISK DATASET
*
*
         L     3,16          CVT
         L     3,0(3)        CVTTCBP
         L     3,4(3)        TCB
         LTR   3,3
         BZ    RACH
         L     3,180(3)      JSCB
         LTR   3,3
         BZ    RACH
         TM    243(3),X'80'  BYPASS PASSWORD PROTECTION FOR THIS JOB ?
         BZ    RACH          NO
         L     3,8(2)        INST. PARM ADDR
         LTR   3,3
         BZ    STOPDEF
         CLC   =C'ARCHIVE',0(3) CALLED BY ONE OF THE ARCHIVE PROGRAMS ?
         BNE   STOPDEF            NO
         L     3,40(2)            NAMING CONVENTIONS ADDRESS
         L     3,36(3)            DATA SET TYPE ADDRESS
         MVI   0(3),X'80'         INDICATE USER DATA SET SO THE ID
* OF THE REQUESTOR WILL NOT BE PLACED IN THE ACCESS LIST OF A
* SPECIFICALLY PROTECTED GROUP DATASET DURING ARCHIVE OPERATIONS
*
         LH    15,=H'12' BPP ARCHIVE PROGRAM ISSUED RACDEF & REQUIRES
         B     RETURN   IT TO BE AUTHORIZED & PROFILE TO BE CREATED
*
RACHTAPE DS    0H
         L     3,8(2)              INSTLN ADDRESS
         C     3,=F'1'             DOES IT CONTAIN JFCB ADDRESS ?
         BE    DEFTAPE             NO - GO CREATE TAPE PROFILE
         LTR   3,3                 WAS THE CALLER RACHECK ?
         BZ    CONTINUE            NO - DON'T CREATE PROFILE
         CLC   0(3,3),21(10)       COMPARE WITH USERID
         BE    DEFTAPE             OK - GO CREATE TAPE PROFILE
         TM    87(3),X'01'         DOES JFCB INDICATE TEMPORARY DS ?
         BO    DEFTAPE             YES - GO CREATE TAPE PROFILE
         B     GETM                NO - GO CHECK AUTHORITY
RACH     L     3,12(2)
         L     4,4(2)    FLAG
         TM    0(4),X'10' NEWNAME?
         BZ    TESTPREF
         L     3,16(2)    NEWNAME ADDR
TESTPREF CLC   0(3,3),21(10)   COMPARE DSN PREF V USERID
         BE    TESTNEW
GETM     GETMAIN RU,LV=WEND-WSTART,SP=0,RELATED=RACH
         LR    8,1
         USING WSTART,8
         MVC   WSTART(WEND-WSTART),RACHECK
DEF      MVC   MODELD(3),0(3)  DS PREF FOR MODEL
         LA    3,MODELD
         IC    7,12(10)   SAVE FLAG FROM ACEE INSTDATA
         RACHECK ENTITY=((3)),VOLSER=DUMMY,ATTR=ALTER,     XXXXXXXXXXXXXX
               MF=(E,(8)),CLASS=DATASET
         STC   7,12(10)   RESTORE ACEE INSTDATA FLAG
         LR    3,15       SAVE RC
         FREEMAIN RU,LV=WEND-WSTART,SP=0,A=(8),RELATED=RACH
         L     4,24(2)             RESOURCE CLASS ADDRESS
         CLC   =C'TAPEVOL',1(4)    TAPE ?
         BE    TESTTAPE            YES
         LTR   3,3
         BNZ   CONTINUE RACDEF WILL FAIL THE RACDEF REQUEST ROUTINELY
*
*
TESTNEW  L     3,4(2)          FLAG
         TM    0(3),X'10'      NEWNAME?
         BNO   STOPDEF         NO
         TM    12(10),X'80'    DOES A PROF EXIST ?
         BO    STOPDEF         NO
         LH    15,=H'12'       YES - ALLOW REQUEST
         L     5,12(10)        ACEEIEP
         LA    5,0(5)
         LTR   5,5
         BZ    RETURN
         L     3,12(2)        DSN ADDR
         MVC   78(44,5),0(3)  SAVE DSN
         L     3,20(2)        VOLSER ADDR
         MVC   122(6,5),0(3)  SAVE VOLSER
         MVI   77(5),X'FF'    INDICATE RACDEF RENAME FOR RACHECK
         B     RETURN
*
*
STOPDEF  LH    15,=H'8'       ADSP OR RENAME WITHOUT PROF - STOP RACDEF
         B     RETURN
*                             PROFILE BEING CREATED,ALLOW DS CREATE.
*
*
*
CONTINUE XR    15,15         RETURN CODE 0
         B     RETURN
*
*
*          DEFINE TAPE
*
TESTTAPE DS    0H
         LTR   3,3            TEST RACHECK RC
         BZ    DEFTAPE        OK - GO DEFINE TAPE
         LH    15,=H'4'       FAIL RACDEF
         B     RETURN
DEFTAPE  CLC   =F'0',8(2)     INST. PARM ADDR
         BE    CONTINUE       NON-ZERO IF RACDEF IN RACHECK POST-EXIT
*
         L     3,44(2)        PROFILE OPTIONS FLAG ADDR
         MVI   0(3),X'04' CAUSE INST. DATA TO BE USED FROM PROFILE
         GETMAIN RU,LV=120,SP=231,RELATED=X  GETMAIN FOR PROFILE
         LR    9,1            ADDRESS OF PROFILE
         MVC   0(4,9),SUBLEN   SUBPOOL, LENGTH
         MVC   4(6,9),=C'XXXXXX'  RESOURCE NAME
         MVI   10(9),C' '
         MVC   11(37,9),10(9)   BLANK OUT REST OF RESOURCE NAME
         MVI   48(9),X'01'      UACC NONE
         MVI   49(9),X'20'      AUDIT FAILURES
         MVC   50(2,9),=H'0'    NONVSAM & LEVEL 0
         MVC   52(4,9),=F'92' VOL SER OFFSET
         MVC   56(4,9),=F'94' ACCESS LIST OFFSET
         MVC   60(8,9),=CL8'TAPEVOL'   CLASS NAME
         MVC   68(4,9),=F'0'
         MVI   68(9),X'10'      GAUDIT NONE
         MVC   72(4,9),=F'105' INST. DATA OFFSET
         MVC   76(4,9),=F'0'
         MVC   80(4,9),=F'0'
         MVC   84(8,9),21(10)
         MVC   92(2,9),=H'0'   NO. OF VOLUME ENTRIES
         MVC   94(2,9),=H'1'  NO. OF ACCESS ENTRIES
         MVC   96(8,9),21(10)  USERID IN ACCESS LIST
         MVI   104(9),X'80'    ALTER AUTH.
         MVC   105(2,9),=H'9'  LENGTH OF INST. DATA
         MVI   107(9),C' '     INST. DATA - INDICATE USE DEFAULT PROF.
         L     1,8(2)          INSTLN ADDRESS
         C     1,=F'1'         IS IT 1 ?
         BE    CREATOR         YES - USE TAPE CREATOR (NO JFCB)
         TM    87(1),X'01'     DOES JFCB INDICATE TEMPORARY DS ?
         BO    CREATOR         YES - USE TAPE CREATOR
         MVC   108(3,9),0(1)   GET DS PREFIX FROM JFCB
         MVC   111(5,9),=C'     ' BLANK REST OF INST DATA
         B     SETADDR
CREATOR  DS    0H
         MVC   108(8,9),21(10) USERID OF TAPE CREATOR
SETADDR  DS    0H
         ST    9,48(2)         STORE ADDR OF PROFILE IN PARM LIST
         LH     15,=H'0'    ACCEPT REQUEST & CONTINUE RACDEF
         B      RETURN
*                               BYPASSING AUTHORITY CHECK
*
*    RACF DELETE OR ADDVOL
*
DELETE   L     3,24(2)        CLASS
         LTR   3,3
         BZ    ABEND5
         CLC   =C'DATASET',1(3)
         BNE   CONTINUE
         TM    12(10),X'80'
         BZ    CHECKPRE         A PROFILE DOES EXIST FOR DATA SET
         LH    15,=H'8'        ALLOW REQUEST BUT STOP SVC PROCESSING
         B     RETURN
*
*
*   CHECK 1ST 3 CHARS. OF DSN VERSUS USERID
*
CHECKPRE L     3,12(2)       DSN ADDR.
         CLC   21(3,10),0(3) COMPARE USERID
         BNE   CONTINUE
         LH    15,=H'12'   ALLOW IF EQUAL
*
*
RETURN   EQU   *
RETURNB  RETURN (14,12),RC=(15)
*
SUBLEN   DC    AL1(231),AL3(116)    SUBPOOL, LENGTH OF PROF.
EXECUTE  EQU   *
ABEND1   EX    0,EXECUTE
ABEND2   EX    0,EXECUTE
ABEND3   EX    0,EXECUTE
ABEND4   EX    0,EXECUTE
ABEND5   EX    0,EXECUTE
DUMMY    DC    CL6'DUMMY '
DATASET  DC    X'07',C'DATASET'
RACHECK  RACHECK MF=L
MODEL    DC    CL44'XXX.RACF.MODEL.PROFILE'
*
WSTART   DSECT
         RACHECK MF=L
MODELD   DC    CL44'XXX.RACF.MODEL.PROFILE'
WEND     EQU   *
         END
```

```
*
*        RACF   COMMAND  PRE-PROCESSING EXIT
*
ICHCNX00 START 0
         SAVE  (14,12),,*
         LR    12,15
         USING ICHCNX00,12
         LR    2,1            PARM LIST ADDR
         L     4,16           CVT
         L     4,0(4)         CVTTCBP
         L     4,12(4)        ASCB
         L     4,108(4)       ASXB
         L     10,200(4)      ACEE
         LTR   10,10
         BZ    CONTINUE      NO ACEE - NOT RACF DEFINED USER
         L     3,28(2)        CLASS
         CLC   =C'DATASET',0(3)
         BNE   CODE
         L     3,12(2)        DSN
         L     4,32(2)        PREFIX
         LTR   4,4
         BZ    CODE
         MVC   0(3,4),1(3)
         MVC   3(5,4),=CL5' ' SET PREFIX = 1ST 3 CHARS. OF DSN
CODE     L     3,4(2)         CALLER CODE ADDR
         LTR   3,3
         BZ    ABEND1
*
*
*    AUTHORIZE  NOSET  COMMANDS
*
         CLC   =X'0302',0(3)
         BE    NOSET          ADDSD NOSET
         CLC   =X'0502',0(3)  DELDSD NOSET ?
         BNE   CONTINUE
*
NOSET    L     3,12(2)
         LTR   3,3
         BZ    ABEND2
         CLC   1(3,3),21(10)  1ST 3 CHARS OF DSN = USERID ?
         BE    AUTH           AUTHORIZE
         CLC   =C'.RACF.MODEL.PROFILE',4(3)   NOSET MODEL DSN
         BNE   GETSTORE
         TM    38(10),X'30'       OPERATIONS OR AUDITOR ?
         BM    AUTH               AUTHORIZE IF EITHER
*
GETSTORE GETMAIN RU,LV=WORKEND-WORKAREA,SP=0,RELATED=CAT
         LR    8,1
         USING WORKAREA,8
         L     4,20(2)        VOL SER LIST ADDR
         LTR   5,4
         BZ    LOCATEA
         CLI   0(5),X'00'     LENGTH 0 ?
         BNE   NOSETB
*
*
*
*
*
*
LOCATEA  MVI   VSAMI,X'00'  INITIALIZE FLAG
```

```
         L     3,12(2)      DSN ADDR
         MVC   DSN,1(3)
LOCATE   MVC   LIST(16),LISTCAT
         LA    3,DSN
         ST    3,LIST+4
         LA    3,WORK
         ST    3,LIST+12
         LOCATE LIST
*
*
*    ANALYZE RC FROM CATALOG SEARCH
*
         LTR   15,15          RC
         BZ    FOUND
         CH    15,=H'4'
         BE    RC4
         CH    15,=H'8'
         BE    RC8
         CH    15,=H'12'
         BE    FREE                  DATASET NOT FOUND
         CH    15,=H'16'
         BE    FREE
         CH    15,=H'20'
         BE    RC20
         CH    15,=H'24'
         BNE   RC28
         TPUT  MSG24,L'MSG24
         B     FREE
RC4      TPUT  MSG4,L'MSG4
         B     FREE
RC8      CH    0,=H'56'
         BE    NOAUTHCT      NO AUTH. TO DO CATALOG SEA
         B     FREE          DS NOT FOUND
NOAUTHCT TPUT  CATP,L'CATP
         B     FREE
RC20     TPUT  MSG20,L'MSG20
         B     FREE
RC28     TPUT  MSG28,L'MSG28
         B     FREE
*
*
*
*
FOUND    EQU   *
         TM    WORK+4,X'20'        DISK ?
         BZ    FREE
         MVC   VOLOB(6),WORK+6
         MVC   LIST(16),LISTOB
         LA    3,DSN
         ST    3,LIST+4
         LA    3,VOLOB
         ST    3,LIST+8
         LA    3,WORKOB
         ST    3,LIST+12
         OBTAIN LIST
         CH    15,=H'4'
         BE    MOUNT
         BL    VTOC
         CH    15,=H'8'
         BE    NODSCB
         TPUT  VTOCIO,L'VTOCIO
         B     FREE
MOUNT    TPUT  MSGMNT,L'MSGMNT
         B     FREE
NODSCB   TPUT  NODS,L'NODS
         B     FREE
*
*
VTOC     TM    WORKOB+39,X'08'     VSAM ?
         BZ    RACH
         MVI   VSAMI,X'FF'       SET FLAG INDICATE VSAM
         MVI   ALIAS,C' '
         MVC   ALIAS+1(43),ALIAS
         CLI   DSN+3,C'.'
         BNE   USER4
         MVC   ALIAS(3),DSN
         B     USERCAT
USER4    CLI   DSN+4,C'.'
         BNE   USER5
         MVC   ALIAS(4),DSN
         B     USERCAT
USER5    CLI   DSN+5,C'.'
         BNE   USER6
         MVC   ALIAS(5),DSN
         B     USERCAT
USER6    CLI   DSN+6,C'.'
         BNE   USER7
         MVC   ALIAS(6),DSN
         B     USERCAT
USER7    CLI   DSN+7,C'.'
         BNE   USER8
         MVC   ALIAS(7),DSN
         B     USERCAT
USER8    MVC   ALIAS(8),DSN
USERCAT  MVC   LIST(16),LISTAL
         LA    3,ALIAS
         ST    3,LIST+4
         LA    3,WORK
         ST    3,LIST+12
         LOCATE LIST
         LTR   15,15
         BZ    RACH         USER CATALOG ALIAS FOUND FOR USERID
         L     4,16      CVT
         L     4,256(4)  AMCBS (AM CONT BLK STRUCTURE)
         L     4,8(4)    MSTR CATS ACB
         L     4,64(4)   CAXWA
         L     4,28(4)   UCB
         MVC   WORK+6(6),28(4) MSTRCTLG VOLSER
*
RACH     LA    5,WORK+5
*
*
*
NOSETB   LA    4,1(5)         1ST VOL SER
         L      3,12(2)
         LA     3,1(3)       DSN
         LA    7,INSTLN   INSTDATA TO PREVENT EXPIRY SIM IN RACHECK
         L     5,28(2)        RESOURCE CLASS ADDR
         LTR   5,5
         BZ    ABEND4
         CLC   =C'DATASET',0(5)
         BNE   CONTINUE       NOT RELEVANT IF NOT DISK DATASET
         LA    5,CLASS
         LA    9,RACHD       LIST FORM ADDR.
         MVC   RACHD(RACHEND-RACHECK),RACHECK
         TM    VSAMI,X'FF'
         BNZ   VSAM                    VSAM
        RACHECK ENTITY=((3)),VOLSER=(4),ATTR=ALTER,MF=(E,(9)),   XXXXXXXX
               CLASS=(5),LOG=NONE,INSTLN=(7)
         B     FREERA
*
*
VSAM    RACHECK ENTITY=((3)),VOLSER=(4),ATTR=ALTER,MF=(E,(9)),   XXXXXXXX
               CLASS=(5),DSTYPE=V,LOG=NONE,INSTLN=(7)
FREERA   LR    3,15       SAVE RACHECK RETURN CODE
         FREEMAIN RU,LV=WORKEND-WORKAREA,SP=0,A=(8),RELATED=CAT
         LTR   3,3
         BNZ   CONTINUE        NO ALTER AUTHORITY - WILL BE REJECTED
*
AUTH     L     3,32(2)       QUALIFIER (PREFIX)
         MVC   0(8,3),21(10) SET QUALIFIER = USERID
         RETURN (14,12),RC=12 GRANT REQUEST & CONTINUE PROCESSING -
FREE     FREEMAIN RU,LV=WORKEND-WORKAREA,SP=0,A=(8),RELATED=CAT
CONTINUE RETURN (14,12),RC=0
RACHECK  RACHECK MF=L
CLASS    DC    X'07',C'DATASET'
RACHEND  EQU   *
TCLASS   DC    X'07',C'TAPEVOL'
ALTER    DC    0H'0',X'0080'
READ     DC    0H'0',X'0002'
INSTLN   DC    C'COMMAND'
EXECUTE  EQU   ABEND1
ABEND1   EX    0,EXECUTE
ABEND2   EX    0,EXECUTE
ABEND4   EX    0,EXECUTE
CATP DC C'NOT AUTHORIZED TO SEARCH CATALOG'
LISTAL   CAMLST NAME,ABEND1,,ABEND1
LISTOB   CAMLST SEARCH,ABEND1,ABEND1,ABEND1
MSGMNT DC C'DATA SET ON UNMOUNTED VOLUME, COMMAND FAILED'
VTOCIO DC C'PERMANENT I/O ERROR IN VTOC OR INVALID DSCB, FAILED'
NODS DC C'DATASET DOES NOT EXIST, ONLY CATLG ENTRY, FAILED'
MSG4     DC    C'CATALOG INACCESSIBLE, UNABLE TO CONTINUE'
MSG20    DC    C'SYNTAX ERROR IN DATASET NAME, UNABLE TO CONTINUE'
MSG24    DC    C'CATALOG ERROR, UNABLE TO CONTINUE'
MSG28    DC    C'UNKNOWN CATALOG ERROR, UNABLE TO CONTINUE'
LISTCAT  CAMLST NAME,ABEND1,,ABEND1
*
*
WORKAREA DSECT
DSN      DC    CL44' '
VOLSER   DC    CL6' '
WORK     DS    0D
         DC    265C' '
VOLOB    DC     CL6' '
WORKOB   DS     0D
         DC     CL140' '
ALIAS    DC     CL44' '
VOLUME   DC     CL6' '
LIST     CAMLST NAME,ABEND1,,ABEND1
VSAMI    DC       X'00'
RACHD    RACHECK MF=L
WORKEND  EQU     *
         END
```

```
*
*                 RACINIT PRE-PROCESSING EXIT
*
ICHRIX01 START 0
         SAVE  (14,12),,*
         LR    12,15
         USING ICHRIX01,12
         LR    2,1            PARMLIST ADDR
         L     3,52(2)        EXIT WORKAREA ADDR
         LTR   3,3
         BZ    ABEND01
         TM    2(3),X'80'     POST-EXIT RETRIED RACINIT + WANTS FAIL ?
         BO    FAIL
         TM    1(3),X'80'
         BO    CONTINUE       RETRY IN PROGRESS
         L     3,4(2)
         LTR   3,3            FLAG ADDR
         BZ    ABENDO
         TM    0(3),X'80'
         BO    DELETE          RACINIT DELETE
         TM    0(3),X'CO'
         BNZ   CONTINUE       NOT CREATE
*
*   CREATE
*
         L     3,8(2)         USERID ADDR
         LTR   3,3
         BZ    ABEND1
         CLI   0(3),X'00'
         BNE   CHECKJOB         USERID WAS SUPPLIED
         L     3,16(2)
         LTR   3,3            PROCNAME ADDR
         BZ    ABEND2
         CLC   =CL8' ',0(3)
         BE    NOSTC          NOT STARTED TASK, NO USERID
*
*   PROMPT OPERATOR FOR USERID + GROUPID OF STARTED TASK
*
         GETMAIN RU,LV=128+WTORE-WTORL,SP=230,RELATED=WTOR
         LR    9,1            REPLY AREA
WTOR     MVI   0(9),C' '
         MVC   1(17,9),0(9)     BLANK OUT REPLY AREA
         LA    6,128(9)     ADDR OF AREA FOR PARM LIST
         MVC   0(WTORE-WTORL,6),WTORL
         LA    8,124(9)       ECB AREA
         XR    3,3
         ST    3,0(8)   CLEAR ECB
        WTOR  ,(9),17,(8),MF=(E,(6))
         WAIT  1,ECB=(8),LONG=YES,RELATED=WTOR
         CLI   0(9),C' '            REPLY BLANK ?
         BE    DEFAULT        ASSIGN CSG USER ,SYS1 GROUP
*
*
         L     3,8(2)         USERID ADDR
         CLI   3(9),C','
         BE    USERA          3 CHAR USERID
         CLI   4(9),C','
         BNE   REPEAT         NOT 3 OR 4 CHAR USERID
         LA    5,5(9)         ADDR OF GROUPID
         MVI   0(3),X'04'     USERID LENGTH
         MVC   1(4,3),0(9)    USERID
         B      GROUPA
USERA    LA     5,4(9)          ADDR OF GROUPID
         MVI    0(3),X'03'      USERID LENGTH
         MVC    1(3,3),0(9)     USERID
GROUPA   CLI    3(5),C' '
         BNE    GROUPB
         LH     7,=H'3'          3 CHAR GROUPID
         B      GROUPD
GROUPB   CLI    4(5),C' '
         BNE    GROUPC
         LH     7,=H'4'         4 CHAR GROUPID
         B      GROUPD
GROUPC   CLI    5(4),C' '
         BNE    REPEAT
         LH     7,=H'5'         5 CHAR GROUPID
GROUPD   L      4,24(2)
         LTR    4,4             GROUPID ADDR
         BZ     ABEND3
         STC    7,0(4)          GROUPID LENGTH
         SH     7,=H'1'         LENGTH NEEDS TO BE ONE LESS FOR MVC
         BM     FREE
         CH     7,=H'7'
         BH     REPEAT
         EX     7,MVCGROUP      GROUPID
         B      FREE
*
*
REPEAT   WTO      'USERID MUST BE 3 OR 4 CHARS. && GROUPID FROM 3 TO 5 CX
                HARS., SEPARATED BY A COMMA',ROUTCDE=(1,2)
         B      WTOR
*                               ASSIGN DEFAULT USER,GROUP FOR STC
DEFAULT  L      3,8(2)          USERID ADDR
         MVC    0(9,3),USER
         L      3,24(2)
         LTR    3,3
         BZ     ABEND4
         MVC    0(9,3),GROUP
FREE     FREEMAIN  RU,LV=128+WTORE-WTORL,SP=230,A=(9),RELATED=WTOR
         L      3,52(2)
         OI     0(3),X'80' WORKAREA RETRY SETTING NO PASSWORD
         B      INSTLN
*
*
*   CHECK JOBNAME 1ST 3 CHARS V. USERID
*
CHECKJOB L      4,16(2)             PROCNAME ADDR
         CLI    0(4),C' '
         BNE    INSTLN                STARTED TASK - DONT CHECK
         L      4,80(2)                JOBNAME ADDR
         LTR    4,4
         BZ     ABEND5
         CLI    0(4),C' '
         BE     INSTLN                 NOT A BATCH JOB SINCE NO JOBNAME
         CLC    0(3,4),1(3)            JOBNAME VERSUS USERID
         BE     INSTLN
         TPUT   MSGA,L'MSGA
         WTO    '1ST 3 CHARS. OF JOBNAME NOT EQUAL TO USERID, JOB FAILEDX
                ',ROUTCDE=(1,2)
         B      FAIL
*
*
```

```
*    NO USERID, NOT STC
*
NOSTC    L     3,80(2)        JOBNAM ADDR
         LTR   3,3
         BZ    ABEND5
         CLI   0(3),C' '
         BE    INSTLN            NO JOBNAME
         L     4,8(2)         USERID ADDR
         MVI   0(4),X'03'     LENGTH
         MVC   1(3,4),0(3)    GET USERID FROM 1ST 3 CHARS. OF JOBNAME
*
*
*
* CODE TO BE INSERTED TO ALLOW RACINIT IN IEFUJV & BYPASS AT JOB START
*
INSTLN   EQU   *
*
*
*
CONTINUE RETURN (14,12),RC=0
*
FAIL     L     3,52(2)
         OI    1(3),X'80'   RETRY INDICATED TO POST EXIT
         RETURN (14,12),RC=4
*
*
DELETE   L     10,32(2)      ACEE ADDR
         LTR   10,10
         BNZ   DELA
         L     10,92(2)      TRY OTHER ACEE PTR
         BZ    CONTINUE      NO ACEE
DELA     XR    4,4
         L     3,12(10)   POINTER TO NEXT GETMAINED AREA
         ST    4,12(10)      CLEAR ACEEIEP TO STOP FREE BY RACF OF
*            OUR AREA IN LSQA, FREED NOW
AGAIN    LA    3,0(3)
         LTR   4,3
         BZ    CONTINUE      NO POINTER, NO MORE AREAS
         L     0,0(4)        SUBPOOL, LENGTH
         L     3,4(4)        POINTER TO NEXT AREA
         FREEMAIN R,LV=(0),A=(4),RELATED=EXPIRY
         B     AGAIN
*
*
EX       EQU   *
ABEND01  EX    0,EX
ABEND0   EX    0,EX
ABEND1   EX    0,EX
ABEND2   EX    0,EX
ABEND3   EX    0,EX
ABEND4   EX    0,EX
ABEND5   EX    0,EX
MSGA     DC    C'1ST 3 CHARS. OF JOBNAME NOT EQUAL TO USERID, JOB FAILEX
               D'
USER     DC    X'03',C'OPS     '
GROUP    DC    X'04',C'SYS1    '
MVCGROUP MVC   1(1,4),0(5)
WTORL    WTOR  'ENTER USERID,GROUPID FOR STC OR RETURN IF NOT NEEDED', X
               ,,,ROUTCDE=(1,2),MF=L
WTORE    EQU   *
         END
```

```
*
*                RACINIT POST-PROCESSING EXIT
*
ICHRIX02 START 0
         SAVE  (14,12),,*
         LR    12,15
         USING ICHRIX02,12
         LR    2,1               PARM LIST ADDR
         L     3,4(2)            FLAG ADDR
         LTR   3,3
         BZ    ABEND0
         TM    0(3),X'C0'
         BNZ   CONTINUE          NOT CREATE
         L     10,32(2)          ACEE ADDR
         LTR   10,10
         BZ    ABEND1
         L     3,52(2)           EXIT WORKAREA ADDR
         LTR   3,3
         BZ    ABEND2
         TM    2(3),X'80'
         BO    CONTINUE          FAIL HAS BEEN SET
         TM    1(3),X'80'
         BO    PASSWD            RETRY IN PROGRESS
         TM    0(3),X'80'
         BO    NOPASS  PASSWORD NOT TO BE CHECKED, SET BY PRE-EXIT
         L     3,16(2)           PROC NAME ADDR
         CLC   =CL8' ',0(3)
         BNE   PASSWD            STARTED TASK, DONT DO ANYTHING
*
*
*   CHECK NOL GROUP - NO PASSWORD & NO BATCH JOBS
*
         CLC   =CL8'NOL',30(10)      NOL GROUP ?
         BNE   TESTSP            CONTINUE NORMALLY
         CLC   =CL8'WMD',21(10)      WMD USER ?
         BE    NOPASS            NO PASSWORD REQD.
         L     3,80(2)           JOBNAME
         CLI   0(3),C' '         BLANK IF TSO USER
         BE    NOPASS            NO PASSWORD FOR TSO USER IN NOL GROUP
         WTO   'NOT ALLOWED TO RUN BATCH JOBS',ROUTCDE=9
         B     FAIL
*
*
TESTSP   TM    38(10),X'80'          SPECIAL ?
         BZ    PASSWD
*
*
*    PROMPT OPERATOR FOR PERMISSION TO RUN JOB OR SESSION BUT NOT STC
*
SPECIAL  GETMAIN RU,LV=128+WTORE-WTORL,SP=230,RELATED=WTOR
         LR    9,1               REPLY AREA
         MVI   0(9),C' '          BLANK REPLY AREA
         LA    8,124(9)          ECB AREA
         XR    3,3
         ST    3,0(8)             CLEAR ECB
         LA    6,128(9)         AREA FOR PARM LIST
         MVC   0(WTORE-WTORL,6),WTORL
         MVC   35(3,6),21(10)        ADD USERID TO MSG
         WTOR  ,(9),10,(8),MF=(E,(6))
         WAIT  1,ECB=(8),LONG=YES,RELATED=WTOR
         IC    3,0(9)             REPLY
         FREEMAIN  RU,LV=128+WTORE-WTORL,SP=230,A=(9),RELATED=WTOR
         N     3,=X'0000003F' STRIP OFF UPPER-LOWER CASE
         CH    3,=X'0024'
         BNE   FAIL              FAIL JOB IF 'U' NOT ENTERED
*
*
         L     3,12(2)           PASSWORD ADDR
         LTR   3,3
         BZ    ABEND3
         CLI   0(3),X'00'
         BNE   PASSWD            PASSWORD IS SUPPLIED
*
*   NO PASSWORD TO BE NEEDED
*
NOPASS   L     3,4(2)            FLAG ADDR
         TM    0(3),X'08'
         BO    PASSWD            NO PASSWORD WAS REQUIRED
         OI    0(3),X'08'        SET NO PASSWORD REQUIRED
         L     3,52(2)           WORKAREA
         MVI   1(3),X'80'        RETRY FLAG FOR EXITS
         B     RETRY             RETRY RACINIT
*
*
*    CHAIN PASSWORD OFF ACEE FOR JOBS TO ACCESS THEIR OWN PASSWORD
*      WHEN SUBMITTING OTHER JOBS TO THE INTERNAL READER.
*    GET LSQA
* BYTES 78 TO 128 ARE USED FOR RENAME COMMANDS WHEN THE OLD DATASET
* HAS A SPECIFIC RACF PROFILE. THE CONTENTS ARE A RENAME FLAG (1 BYTE)
* THE OLD DSN (44 BYTES) AND THE VOLSER (6 BYTES)
*
PASSWD   GETMAIN RU,LV=128,SP=235,RELATED=PASSWORD  STORE PASSWORD
         L     3,28(2)             NEW PASSWORD ADDR
         LTR   3,3
         BZ    OLDPASS
         CLI   0(3),X'00'
         BNE   PASS              USE NEW PASSWORD
OLDPASS  L     3,12(2)           PASSWORD ADDR
         LTR   3,3
         BZ    ABEND4
PASS     MVC   0(4,1),SPLEN       SUBPOOL & LENGTH
         MVC   8(9,1),0(3)       PASSWORD
         MVI   17(1),C' '
         MVC   18(54,1),17(1)      BLANK REST OF AREA
         XR    4,4
         ST    4,4(1)              ZERO POINTER TO NEXT AREA
         STCM  1,7,13(10)        POINT TO PASSWORD FROM ACEEIEP
*
*
*
CONTINUE RETURN (14,12),RC=0
*
FAIL     L     3,52(2)
         MVI   2(3),X'80'        FAIL ON RETRY
RETRY    RETURN (14,12),RC=4
*
*
SPLEN    DC    AL1(235),AL3(72)
WTORL    WTOR  'REPLY U TO ALLOW USER  XXX WITH SPECIAL AUTHORITY TO COX
               NTINUE, REPLY ANY OTHER CHARACTER TO CANCEL',           X
               ROUTCDE=(1,2),MF=L
WTORE    EQU   *
```

```
EX        EQU    *
ABEND0    EX     0,EX
ABEND1    EX     0,EX
ABEND2    EX     0,EX
ABEND3    EX     0,EX
ABEND4    EX     0,EX
          END
```

```
*        RACHECK PRE-PROCESSING EXIT
*         UPDATED BY JCG 6/12/79
*
ICHRCX01 START 0
         SAVE  (14,12),,*
         LR    12,15
         USING ICHRCX01,12
         LR    2,1            RACHECK EXIT PARM LIST ADDR
         L     4,16           CVT
         L     4,0(4)         CVTTCBP
         L     4,12(4)        ASCB
         L     4,108(4)       ASXB
         L     10,200(4)      ACEE
         LTR   10,10
         BZ    CONTINUE      NO ACEE - NOT RACF DEFINED USER
         L     3,36(2)      WORKAREA FOR RACHECK EXITS ADDR
         LTR   3,3
         BZ    ABEND2
         TM    1(3),X'80'      RETRY WITH MODEL PROFILE?
         BNZ   CONTINUE        - BYPASS EXIT IF RETRY
         NI    12(10),X'7F'    ZERO NO PROF. BIT SET BY POST-EXIT ANTE
*
*
* TEST FOR PECULIAR USERS
*
         CLC   =CL8'NOL',30(10)     GROUP NOL ?
         BNE   XTN
         CLC   =CL8'SUP',21(10)     USER SUP ?
         BNE   RESTRICT
         L     3,8(2)                FLAG
         TM    0(3),X'FC'            GREATER THAN READ REQD.
         BM    SUPFAILA              NOT ALLOWED FOR SUP
         L     3,24(2)               CLASS
         CLC   =C'DATASET',1(3)      DISK DATASET ?
         BNE   NOLFAILA              NO TAPE ALLOWED
         L     3,20(2)               DSN
         CLC   =C'ADP',0(3)          ADP = PREFIX ?
         BE    CLASS                 ALLOW IF ADP PERMITS
RESTRICT L     3,24(2)
         CLC   =C'DATASET',1(3)
         BNE   NOLFAILA
         L     3,20(2)
         CLC   21(3,10),0(3)         USERID=PREFIX ?
         BE    CLASS                 ALLOW
         CLC   =C'SYS',0(3)          SYS=PREFIX ?
         BE    CLASS                 ALLOW IF SYS PERMITS ?
        CLC   =C'USE',0(3)       USE=PREFIX ?
        BE    CLASS              ALLOW IF USE PERMITS
        CLC   =C'RFD',0(3)       **** ALLOW ACCESS TO RFD IF HE
        BE    CLASS              PERMITS - TEMPORARY ONLY ****
        B     NOLFAILB
XTN     CLC   =CL8'XTN',30(10)    GROUP XTN ?
        BE    RESTRICT            RESTRICT TO OWN & SYSTEM DATASETS
CLASS    L     3,24(2)        RESOURCE CLASS TO BE CHECKED
         LTR   3,3
         BZ    ABEND3
         CLC   =C'DATASET',1(3)
         BNE   CONTINUE
*
*                             DISK DATASET
         L     11,60(2)       COMMAND EXIT PARM LIST ADDR.
         LTR   11,11
         BZ    ABEND1
         L     3,32(11)      QUALIFIER ADDR
         MVC   3(5,3),=CL5' ' MAKE SURE PREFIX IS JUST 1ST 3 CHS.
*
*         TEST FOR FASTPATH
*
*
TESTFAST L     3,32(11)       DSN PREFIX ADDR.
         LTR   3,3
         BZ    ABEND4
         CLC   21(3,10),0(3)  USERID FROM ACEE VERSUS DSN PREFIX
         BNE   SIMULATE
         L     3,4(2)        FLAG BYTE 1 ADDR
         TM    0(3),X'01'    (ENTITY,CSA) ?
         BO    AVOID         AVOID FASTPATH IF CSA
         L     3,8(2)       FLAG2 ADDR
         TM    0(3),X'80'    ALTER AUTH REQD
         BZ    FASTPATH   AVOID FASTPATH IF ALTER
* (IN DELETE RACDEF EXIT NEEDS TO KNOW IF PROF. EXISTS
* - RACHECK EXITS TELL IT IF NOT FASTPATH).
AVOID    L     3,36(2)          USER FLAGS ADDR
         OI    2(3),X'80'       FASTPATH AVOIDED
         L     3,32(11)         QUALIFIER ADDR
         MVC   0(8,3),=CL8' '   PREVENT SVC FASTPATH
*
*
*
SIMULATE L     3,8(2)         ACCESS REQUESTED FLAG ADDR
         TM    0(3),X'02'      READ ?
         BO    READ
*           UPDATE, CONTROL OR ALTER REQUESTED
         L     3,20(2)         ENTITY ADDR
         CLC   =C'SYS1',0(3)    SYS1  ?
         BE    EXPIRY          SIMULATE DATE PROTECT
         CLC   =C'IMS1',0(3)
         BE    EXPIRY
         CLC   =C'USER',0(3)
         BE    EXPIRY
         B     GDG
READ     L     3,20(2)
         CLC   =C'SYS1.OPSAUTH',0(3)
         BE    EXPIRY           SIMULATE PASSWORD READ PROTECT
         CLC   =C'SYS1.RACF',0(3)
         BE    EXPIRY
*
*
         CLC   =C'SYS1.FORTLIB',0(3)
         BE    FASTPATH
         CLC   =C'SYS1.CLIST',0(3) FASTPATH FOR COMMONLY USED
         BE    FASTPATH            SYSTEM DATASETS
         CLC   =C'USER.CLIST',0(3)
         BE    FASTPATH
         CLC   =C'SYS1.PLIBASE',0(3)
         BE    FASTPATH
         CLC   =C'SYS1.COBLIB',0(3)
         BE    FASTPATH
         CLC   =C'SYS1.BASICLIB',0(3)
         BE    FASTPATH
         B     GDG
*
```

```
*
* SIMULATE EXPIRY DATE PROTECT OR READ PROTECT
EXPIRY   L     4,16(2)        INSTDATA
         LTR   4,4
         BZ    EXPIRYA
         CLC   =C'COMMAND',0(4) CALLED FROM CATFIND OR COMMAND EXIT
         BE    GDG           AVOID EXPIRY DATE AUTH. IF FROM COMMAND
EXPIRYA  L     4,4(2)         FLAG BYTE 1 ADDR
         TM    0(4),X'10'     VSAM ?
         BNZ   GDG     DONT SIMULATE EXPIRY DATE PROTECT FOR VSAM DSETS
         L     4,12(10)      ACEEIEP
         LA    4,0(4)
         LTR   4,4
         BZ    GDG NO PTR TO PASSWORD - ONLY STCS CAN SKIP WTOR
REPEAT   L     5,4(4)
         LTR   5,5
         BZ    NOTFOUND
         CLC   0(44,3),8(5)
         BE    GDG
         LR    4,5
         B     REPEAT
*
NOTFOUND TPUT  OPER,L'OPER
REASK    GETMAIN RU,LV=128+WTORE-WTORL,SP=230,RELATED=WTOR
         LR    9,1        REPLY AREA
         MVI   0(9),C' '
         LA    8,124(9)   ECB AREA
         XR    3,3
         ST    3,0(8)     CLEAR ECB
         LA    6,128(9)   AREA FOR PARMLIST
         MVC   0(WTORE-WTORL,6),WTORL
         L     3,20(2)
         MVC   79(44,6),0(3)        DSN
         L     3,28(2)
         MVC   60(6,6),0(3)         VOLSER
         L     3,16         CVT
         L     3,0(3)       CVTTCBP
         L     3,4(3)       CURRENT TCB
         L     3,12(3)      TIOT
         CLI   16(3),C' '   PROC CALLING STEPNAME ?
         BE    MOVESTEP     NO PROCEDURE
         LA    3,8(3)       USE CALLING STEPNAME
MOVESTEP MVC   34(7,6),8(3) MOVE STEPNAME(6 CH) INTO WTO MESSAGE
         MVC   29(3,6),21(10)       USERID
         WTOR  ,(9),10,(8),MF=(E,(6))
         WAIT  1,ECB=(8),LONG=YES,RELATED=WTOR
         IC    3,0(9)
         FREEMAIN RU,LV=128+WTORE-WTORL,SP=230,A=(9),RELATED=WTOR
         N     3,=X'0000003F'   STRIP OFF UPPERCASE
         CH    3,=X'0024'       'U' ?
         BE   APPROVE
*
         CH   3,=X'0014'      'M'
         BE   FAIL
         WTO  'REPLY "U" TO ALLOW ACCESS, "M" TO REFUSE ACCESS', XXXXXXX
               ROUTCDE=(1,2,11)
         B     REASK
*
*
APPROVE  GETMAIN RU,LV=56,SP=241,RELATED=EXPIRY   CSA
         ST    1,4(4)               CHAIN TO PREVIOUS AREA
         MVC   0(4,1),SUBLEN          SUBPOOL, LENGTH
         MVC   4(4,1),=F'0'           ZERO PTR TO NEXT AREA(DOESNT EXIST)
         L     3,20(2)
         MVC   8(44,1),0(3)           STORE DSN SO ONLY ONE OP. REPLY
*                                     FOR EACH DATASET.
*
*
*
*
*               IS IT A GDG ?
*
GDG      L     3,12(11)           DSN ADDR
         LTR   3,3
         BZ    ABEND5
         XR    4,4
         IC    4,0(3)             DSN LENGTH
         LTR   4,4
         BZ    ABEND6
         SH    4,=H'7'
         BNP   CONTINUE
         AR    3,4                1ST CHAR. OF GDG IDENT. (IF PRESENT)
         CLI   0(3),C'G'
         BNE   CONTINUE
         CLI   5(3),C'V'
         BNE   CONTINUE
         TM    1(3),X'F0'         NUMERIC ?
         BNO   CONTINUE
         TM    2(3),X'F0'
         BNO   CONTINUE
         TM    3(3),X'F0'
         BNO   CONTINUE
         TM    4(3),X'F0'
         BNO   CONTINUE
         TM    6(3),X'F0'
         BNO   CONTINUE
         TM    7(3),X'F0'
         BNO   CONTINUE
*
*          GDG - SET DSN=GDG BASE NAME
*
         L     3,12(11)           DSN ADDR
         LTR   3,3
         BZ    ABEND7
         XR    4,4
         IC    4,0(3)             DSN LENGTH
         SH    4,=H'9'            NEW DSN LENGTH
         L     3,20(2)            DSN ADDR. IN RACHECK EXIT PARM LIST
         LTR   3,3
         BZ    ABEND8
         AR    3,4
         MVC   0(9,3),=CL9' ' BLANK OUT .GNNNNVNN
         L     3,56(2)           OLDVOL ADDR
         LTR   3,3
         BZ    GDGA
         MVC   0(6,3),=CL6' ' BLANK OUT OLDVOL
GDGA     L     3,28(2)           VOLSER ADDR
         LTR   3,3
         BZ    ABEND9
         MVC   0(6,3),=C'DUMMY ' VOL SER OF MODEL PROFILES
         OI    12(10),X'80' SET NOPROF. - CAN ONLY EXIST FOR GDGBASE
*
```

```
*
CONTINUE RETURN (14,12),RC=0
*
FASTPATH L     3,36(2)        RACHECK EXIT WORKAREA ADDR.
         MVI   0(3),X'80'     TELL POST RACHECK EXIT TO ALLOW ACCESS
*                             CAUSE POST-PROC. EXIT BYPASS.
RETURNB  RETURN (14,12),RC=8  BYPASS RACHECK
FAIL WTO 'ACCESS TO THE DATASET HAS BEEN REFUSED BY THE OPERATOR', XXXXX
               ROUTCDE=(1,2,11)
FAILURE  L     3,36(2)          WORKAREA
         OI    0(3),X'80' STOP POSTEXIT RETRY BY FLAG ACCESS ALLOWED
         RETURN (14,12),RC=4      FAIL ACCESS REQUEST
SUPFAILA WTO 'SUP NOT ALLOWED MORE THAN READ ACCESS',ROUTCDE=(9,11)
         B     FAILURE
NOLFAILA WTO 'NO ACCESS TO MAGNETIC TAPE IS ALLOWED',ROUTCDE=(9,11)
         B     FAILURE
NOLFAILB WTO 'ACCESS TO DATA SET NOT ALLOWED - NOT SYSTEM OR OWN', XXXXX
               ROUTCDE=(9,11)
         B     FAILURE
SUBLEN   DC    AL1(241),AL3(56)
WTORL WTOR      'REPLY U TO ALLOW XXX (XXXXXXX) ACCESS ON VOLUME XXXXXXXZ
                TO DATA SET                                            Z
                 ',                                                    Z
               ROUTCDE=(1,2),MF=L
WTORE    EQU   *
OPER DC C'OPERATOR AUTHORIZATION IS NEEDED TO MODIFY THE DATASET'
         DS    0H
EXECUTE  EQU   *
ABEND1   EX    0,EXECUTE
ABEND2   EX    0,EXECUTE
ABEND3   EX    0,EXECUTE
ABEND4   EX    0,EXECUTE
ABEND5   EX    0,EXECUTE
ABEND6   EX    0,EXECUTE
ABEND7   EX    0,EXECUTE
ABEND8   EX    0,EXECUTE
ABEND9   EX    0,EXECUTE
         END
```

```
*
*              RACHECK POST-PROCESSING EXIT
*
ICHRCX02 START 0
         SAVE  (14,12),,*
         LR    12,15
         USING ICHRCX02,12
         LR    2,1            RACHECK EXIT PARM LIST ADDR.
         L     4,16           CVT
         L     4,0(4)         CVTTCBP
         L     4,12(4)        ASCB
         L     4,108(4)       ASXB
         L     10,200(4)      ACEE
         XR    15,15        RC IF NO ACEE
         LTR   10,10
         BZ    RETURNB      NO ACEE - NOT RACF DEFINED USER
         L     3,36(2)        RACHECK EXIT WORKAREA ADDR.
         LTR   3,3
         BZ    ABEND2
         TM    0(3),X'80'
         BO    CONTINUE         PRE-PROC. EXIT ALLOWED ACCESS
         TM    1(3),X'80'
         BZ    RACHECK      RACHECK WAS NOT REPEATED USING MODEL PROFILE
*
         L     3,20(2)      RESOURCE ADDR
         L     5,12(10)     ACEEIEP
         LA    5,0(5)
         LTR   5,5
         BZ    CONTINUE
         MVC   0(44,3),20(5)  RESTORE DSN OR VOLSER, GET RID OF MODEL
         L     3,24(2)        CLASS ADDR
         MVC   1(7,3),64(5)   RESTORE CLASS
         MVI   0(3),X'07'
         L     3,28(2)        VOLSER ADDRESS
         MVC   0(6,3),71(5)   RESTORE VOLSER
         B     CONTINUE
*
*
*   RACHECK MACRO WAS THE ORIGINAL CALLER OF RACHECK
*
RACHECK  L     3,48(2)        ACCESS CODE ADDR
         LTR   3,3
         BZ    ABEND5
         CLI   0(3),X'00'
         BE    NOPROF         NO PROFILE WAS FOUND IF CODE=0
         L     3,24(2)        CLASS ADDR.
         LTR   3,3
         BZ    ABEND6
         CLC   =C'TAPEVOL',1(3)
         BNE   CONTINUE       ALLOW RACHECK TO VERIFY ACCESS IF ¬ TAPE
*
*  TAPE
         L     3,32(2)        INSTALLATION DATA ADDR FROM TAPE PROFILE
         LTR   3,3
         BZ    ALLOW
TAPEA    CLC   2(3,3),21(10)  COMPARE OWNER OF TAPE WIT
         BE    ALLOW          ALLOW ACCESS IF USER IS CREATOR OF TAPE
         CLI   1(3),C' '
         BNE   CONTINUE       SPECIFIC AUTHORITY DEFINED ON TAPE PROF.
         L     5,24(2)        CLASS ADDR
         LTR   5,5
         BZ    ABEND7
         MVC   0(8,5),DATASET CHANGE CLASS TO DATASET
         L     4,20(2)        RESOURCE ADDR
         LTR   4,4
         BZ    ABEND8
         L     5,12(10)       ACEEIEP
         LA    5,0(5)
         LTR   5,5
         BZ    NOSAVEA
         MVC   20(6,5),0(4)    SAVE VOLSER
         MVI   26(5),C' '
         MVC   27(37,5),26(5)
         MVC   64(7,5),=C'TAPEVOL'   SAVE CLASS
NOSAVEA  MVC   0(3,4),2(3)    MOVE OWNER INTO PREFIX
         LA    4,3(4)         TAPE OWNER IS 3 CHAR. USERID
         B     MODELB
*
*          NO PROFILE FOUND
*
NOPROF   L     3,24(2)        CLASS ADDR
         LTR   3,3
         BZ    ABEND9
         CLC   =C'TAPEVOL',1(3)
         BE    TAPEDEF        DEFINE PROFILE FOR TAPE VOLUME
         CLC   =C'DATASET',1(3)
         BNE   CONTINUE
*
*  USE MODEL IF NO DISK PROFILE OR NO SPECIFIC PROT. IN TAPE PROFILE
*
         L     4,20(2)        DSN ADDR
         LTR   4,4
         BZ    ABEND10
         L     5,12(10)      ACEEIEP
         LA    5,0(5)
         LTR   5,5
         BZ    NOSAVEB
*
* BYPASS THE RETRY WITH THE MODEL IF THIS IS PART OF A RENAME
*
         CLI   77(5),X'FF'      RACDEF RENAME ?
         BNE   SAVEDSN          NO
         CLC   78(44,5),0(4)    SAME DSN ?
         BNE   SAVEDSN          NO
         L     3,28(2)          VOLSER ADDR
         CLC   122(6,5),0(3)    SAME VOLSER ?
         BNE   SAVEDSN          NO
         MVI   77(5),X'00'
         B     CONTINUE         YES - RETURN WITH 'PROF NOT FOUND'
SAVEDSN  MVC   20(44,5),0(4)    SAVE DSN
         MVC   64(7,5),=C'DATASET'  SAVE CLASS
         L     3,28(2)              VOLSER ADDRESS
         MVC   71(6,5),0(3)         SAVE VOLSER
NOSAVEB  LA    4,3(4)               3 OR 4 CHAR PREFIXES
MODELB   L     3,20(2)
         MVI   20(3),C' '
         MVC   21(24,3),20(3) BLANK DSN
         MVC   0(19,4),=C'.RACF.MODEL.PROFILE' MODEL DSN
         L     3,56(2)     OLDVOL ADDR
         LTR   3,3
         BZ    MODELA
         MVC   0(6,3),=CL6' ' BLANK OUT OLDVOL
```

```
MODELA   L     3,28(2)       VOLSER ADDR
         LTR   3,3
         BZ    ABEND11
         MVC   0(6,3),=C'DUMMY ' VOLSER OF DEFAULT PROFILES
         L     3,4(2)           FLAG1 ADDR
         LTR   3,3
         BZ    ABEND12
         NI    0(3),X'EF'       SET DSTYPE =NONVSAM
         L     3,36(2)          WORKAREA ADDR
         MVI   1(3),X'80'       INDICATE RACHECK RETRY TO EXITS
         OI    12(10),X'80' TELL RACDEF NO PROFILE FOUND - MODEL USED
         LH    15,=H'4'         RETURN CODE
         B     RETURN
*
*
*   ISSUE RACDEF FOR TAPE VOLUME
*
TAPEDEF  TM    38(10),X'01' ACEE USER FLAGS - RACF DEFINED USER ?
         BZ    CONTINUE     DONT DEFINE TAPE PROF IF NOT RACF USER
         L     3,20(2)         VOLUME SERIAL NO. ADDR (ENTITY ADDR)
         LTR   3,3
         BZ    ABEND13
         CLI   0(3),C'9'  ONLY CREATE PROFILE FOR 9XXXXX SERIES VOLS
         BNE   CONTINUE
         CLI   5(3),C' '
         BE    CONTINUE
         GETMAIN RU,LV=32,SP=0,RELATED=RACDEF
         LR    9,1
         MVC   0(32,9),RACDEF
         L     4,16(2)               ADDRESS OF INSTALLATION PARM
         LTR   4,4                   ANY SPECIFIED ?
         BNZ   LEAVE4                YES - CONTAINS ADDRESS OF JFCB
         LA    4,1                   NO - JUST INDICATE RACDEF CALLED
*                                    FROM HERE BY NONZERO INSTLN FIELD
LEAVE4   DS    0H
         RACDEF ENTITY=(3),TYPE=DEFINE,INSTLN=(4),MF=(E,(9))
         LR    3,15       SAVE RACDEF RETURN CODE
         FREEMAIN RU,LV=32,SP=0,A=(9),RELATED=RACDEF
         LTR   3,3
         BZ    ALLOW        RACDEF SUCCEEDED
         TPUT  MSG,L'MSG
 WTO 'USER DOES NOT HAVE AUTHORITY TO DEFINE TAPE DATA SET',     XXXXXXX
               ROUTCDE=(1,2,11)
 WTO 'ALTER AUTHORITY REQUIRED IN DEFAULT RACF PROFILE OF OWNER',     XX
               ROUTCDE=(1,2,11)
         ABEND 2323,,STEP,SYSTEM  ABEND 913
*
*
*   ALLOW ACCESS
*
ALLOW    L     8,40(2)          ABEND CODE ADDR
         LTR   8,8
         BZ    ABEND14
         XR    3,3
         ST    3,0(8)
         L     9,44(2)          RETURN CODE ADDR.
         LTR   9,9
         BZ    ABEND15
         ST    3,0(9)
         L     3,48(2)      ACCESS CODE ADDR.
         MVI   0(3),X'80'   ALTER AUTH.
```

```
*
*
*
CONTINUE XR    15,15      RETURN CODE 0
*
RETURN   L     3,24(2)       CLASS ADDR.
         CLC   =C'DATASET',1(3)
         BNE   RETURNB
         L     3,20(2)        DSN ADDR.
         CLC   21(3,10),0(3)  USERID VERSUS 1ST 3CHARS. OF DSN
         BNE   RETURNB
         XR    4,4
         L     3,40(2)       CLEAR RC & ABENDCODE
         ST    4,0(3)
         L     3,44(2)
         ST    4,0(3)
         L     3,48(2)        ACCESS CODE ADDR
         MVI   0(3),X'80'     ALTER AUTH.
         L     3,60(2)       COMMAND PARMLIST
         L     3,32(3)       QUALIFIER ADDR
         MVC   0(8,3),21(10) PLACE USERID IN QUALIFIER
         XR    15,15          AVOID RETRY IF USERID=DSN PREFIX
RETURNB  RETURN (14,12),RC=(15)
*
*
*
DATASET  DC    X'07',C'DATASET'
MSG     DC    C'USER NOT ALLOWED TO DEFINE TAPE VOLUME'
RACDEF  RACDEF MF=L,CLASS='TAPEVOL'
EXECUTE  EQU   *
ABEND1   EX    0,EXECUTE
ABEND2   EX    0,EXECUTE
ABEND3   EX    0,EXECUTE
ABEND4   EX    0,EXECUTE
ABEND5   EX    0,EXECUTE
ABEND6   EX    0,EXECUTE
ABEND7   EX    0,EXECUTE
ABEND8   EX    0,EXECUTE
ABEND9   EX    0,EXECUTE
ABEND10  EX    0,EXECUTE
ABEND11  EX    0,EXECUTE
ABEND12  EX    0,EXECUTE
ABEND13  EX    0,EXECUTE
ABEND14  EX    0,EXECUTE
ABEND15  EX    0,EXECUTE
         END
```

```
//JLR JOB ,,CLASS=X,MSGCLASS=A,NOTIFY=JLR
//SC     EXEC ASMFCL,MAC1='DLIB.AMODGEN',PARM.LKED='AC=1,LET,LIST,MAP'
//ASM.SYSPRINT DD SYSOUT=*
//ASM.SYSIN DD *
CATFIND  START 0
         SAVE  (14,12),,*
         LR    12,15
         USING CATFIND,12
         LA    9,4092(12)
         USING CATFIND+4092,9
         ST    13,SAVE+4
         LR    11,13
         LA    13,SAVE
         ST    13,8(11)
         LR    11,1
         USING CPPL,11
         L     3,CPPLCBUF
         ST    3,PPLCOM
         L     3,CPPLUPT
         ST    3,PPLUPT
         L     3,CPPLECT
         ST    3,PPLECT
         XC    ECB,ECB
         CALLTSSR EP=IKJPARS,MF=(E,PPL)
         L     10,ANS
         USING IKJPARMD,10
         LTR   15,15
         BZ    CONTINUE
         LA    1,GFPOINTR
         ST    15,GFRCODE
         LA    3,GFPARSE
         STH   3,GFCALLID
         ST    2,GFCPPLP
         LA    3,PROGNAME
         ST    3,GFPGMNP
         LINK  EP=IKJEFF19
CONTINUE EQU *
         TM    AUTHB+6,X'80'        AUTH PARM CODED
         BZ    READ
         L     3,AUTHB
         CLC   =C'READ',0(3)
         BE    READ
         CLC   =C'UPDA',0(3)
         BE    UPDATE
         CLC   =C'CONT',0(3)
         BE    CONTROL
         CLC   =C'ALTE',0(3)
         BE    ALTER
         TPUT  AUTHMSG,L'AUTHMSG
         B     EXIT
READ     MVI   ACCESS,X'02'
         B     DSNAA
UPDATE   MVI   ACCESS,X'04'
         B     DSNAA
CONTROL  MVI   ACCESS,X'08'
         B     DSNAA
ALTER    MVI   ACCESS,X'80'
DSNAA    L     2,DSNM               DSN ADDR
         LH    3,DSNM+4            DSN LEN
         SH    3,=H'1'
         EX    3,MVCD
         TM    GENB+6,X'80'
         BZ    TESTVOL
         LA    4,DSN          DSN ADDR
         AR    4,3            ADD DSN LENGTH-1
         LA    4,1(4)         ADD 1
         L     2,GENB         ADDR OF GENERATION
         CLI   0(2),C'+'      + GENERATION ?
         BNE   NEGZERO
         MVC   0(3,4),=C'(0)' RESET TO CURRENT GENERATION
         B     TESTVOL
NEGZERO  LH    3,GENB+4     LENGTH
         MVI   0(4),C'('
         EX    3,MVCG    ADD GENERATION NO. TO DSN
         AR    4,3     LEN
         MVI   1(4),C')'
TESTVOL  EQU   *
         TM    VOLB+6,X'80'
         BZ    LOCATE
         L     3,VOLB
         LH    4,VOLB+4           LENGTH
         SH    4,=H'1'
         EX    4,MVCVOL           MOVE VOLSER
         CLC   =C'ARCHIV',0(3)
         BE    ARCHIVE
         B     RDISK     NO NEED TO SEARCH CATALOG IF VOLSER CODED
LOCATE   LOCATE LIST
*
*
*     ANALYZE RC FROM CATALOG SEARCH
*
         LTR   15,15          RC
         BZ    FOUND
         CH    15,=H'4'
         BE    RC4
         CH    15,=H'8'
         BE    RC8
         CH    15,=H'12'
         BE    ARCHIVE               DATASET NOT FOUND
         CH    15,=H'16'
         BE    ARCHIVE
         CH    15,=H'20'
         BE    RC20
         CH    15,=H'24'
         BNE   RC28
         TPUT  MSG24,L'MSG24
         B     EXIT
RC4      TPUT  MSG4,L'MSG4
         B     EXIT
RC8      CH    0,=H'56'
         BE    NOAUTHCT       NO AUTH. TO DO CATALOG SEA
         B     ARCHIVE        DS NOT FOUND
NOAUTHCT TPUT  CATP,L'CATP
         B     EXIT
RC20     TPUT  MSG20,L'MSG20
         B     EXIT
RC28     TPUT  MSG28,L'MSG28
         B     EXIT
*
*
ARCHIVE  MVC   WORK+6(6),=C'ARCHIV'
*
```

```
          CALL  SHRCAT      ENQ SHR ON ARCHIVE CAT
          OPEN  (CAT)       OPEN ARCHIVE CAT
          LTR   15,15
          BNZ   CATOPERR    ERROR
          GET   RPL=ARCH    READ RECORD FROM  CAT
          LTR   3,15
          BNZ   GETRC          ERROR
          L     15,RECADDR   GET ADDR OF DATA RECORD
          TM    0(15),X'04'  TEST VSAM BIT
          BZ    CLOSECAT     NOT VSAM
          MVC   LISTI+10(4),=C'VSAM' INDICATE VSAM DS, NVSAM PROF.
          B     CLOSECAT
GETRC     SHOWCB RPL=ARCH,AREA=ARCHRC,LENGTH=4,FIELDS=(FDBK) GET RC
CLOSECAT  CLOSE (CAT)       CLOSE ARCHIVE CAT
          CALL  DEQCAT      FREE ARCHIVE CAT
          LTR   3,3
          BZ    RDISK       DSN FOUND IN ARCHIVE CAT
          CH    3,=H'12'
          BE    CATPHERR    PHYSICAL ERROR
          L     15,ARCHRC   GET RC
          CH    15,=H'16'   RECORD NOT FOUND
          BNE   CATLOERR    NO - LOGICAL ERROR
          TPUT  NODSN,L'NODSN    DSN NOT FOUND
          B     EXIT
CATPHERR  TPUT  ARCHPH,L'ARCHPH
          B     EXIT
CATLOERR  TPUT  ARCHLO,L'ARCHLO
          B     EXIT
CATOPERR  TPUT  ARCHOP,L'ARCHOP
          CALL  DEQCAT
          B     EXIT
          B     RDISK
*
*
FOUND     EQU   *
          TM    WORK+4,X'20'        DISK ?
          BZ    RACH
          MVC   VOLOB(6),WORK+6
          OBTAIN LISTOB
          CH    15,=H'4'
          BE    MOUNT
          BL    VTOC
          CH    15,=H'8'
          BE    NODSCB
          TPUT  VTOCIO,L'VTOCIO
          B     EXIT
MOUNT     TPUT  MSGMNT,L'MSGMNT
          B     EXIT
NODSCB    TPUT  NODS,L'NODS
          B     EXIT
*
*
VTOC      TM    WORKOB+39,X'08'     VSAM ?
          BZ    RACH
          MVI   VSAMI,X'FF'       SET FLAG INDICATE VSAM FOR RACHECK
          MVC   LISTI+10(4),=C'VSAM'
          CLI   DSN+3,C'.'
          BNE   USER4
          MVC   ALIAS(3),DSN
          B     USERCAT
USER4     CLI   DSN+4,C'.'
```

```
         BNE   USER5
         MVC   ALIAS(4),DSN
         B     USERCAT
USER5    CLI   DSN+5,C'.'
         BNE   USER6
         MVC   ALIAS(5),DSN
         B     USERCAT
USER6    CLI   DSN+6,C'.'
         BNE   USER7
         MVC   ALIAS(6),DSN
         B     USERCAT
USER7    CLI   DSN+7,C'.'
         BNE   USER8
         MVC   ALIAS(7),DSN
         B     USERCAT
USER8    MVC   ALIAS(8),DSN
USERCAT  LOCATE LISTAL
         LTR   15,15
         BZ    RACHAA         USER CATALOG ALIAS FOUND FOR USERID
         CALL  MCATVOL,(VOLUME),VL
         MVC   WORK(2),=H'1'  NO. OF ENTRIES
         MVC   WORK+6(6),VOLUME      MASTER CATALOG VOLUME
RACHAA   MVI   WORK+4,X'20'         INDICATE DISK DATASET
*
RACH     LH    3,WORK      NO. OF ENTRIES
         LA    4,WORK+6       VOLUME ENTRY
         LA    5,LISTC+9      CLIST CMMND TO BE BUILT
LOOPVOL  MVC   0(6,5),0(4)    MOVE VOLSER
         LA    4,12(4)        INCREMENT ENTRY
         LA    5,6(5)         INCREMENT DESTINATION
         BCT   3,LOOPVOL      LOOP UNTIL FINISHED
         TM    WORK+4,X'80'    TAPE ?
         BZ    DISK
         MVC   LISTD+10(4),=C'TAPE' MOVE INTO COMMAND SET &UNIT=
         XR    3,3
         IC    3,ACCESS
         MVC   RESOURCE(6),WORK+6
         RACHECK ENTITY=(RESOURCE,CSA),CLASS='TAPEVOL',ATTR=(3),  XXXXXX
               LOG=NONE
         B     ANALYZE
DISK     MVC   LISTD+10(4),=C'DISK'
RDISK    XR    3,3
         IC    3,ACCESS
         MVC   VOLSER,WORK+6
         TM    VSAMI,X'FF'
         BZ    NONVSAM
         RACHECK ENTITY=(DSN,CSA),CLASS='DATASET',ATTR=(3),     XXXXXXXX
               VOLSER=VOLSER,DSTYPE=V,LOG=NONE,INSTLN=INSTLN
         B     ANALYZE
NONVSAM  RACHECK ENTITY=(DSN,CSA),CLASS='DATASET',ATTR=(3),     XXXXXXXX
               VOLSER=VOLSER,LOG=NONE,INSTLN=INSTLN
*
*
*     ANALYZE RESULT OF RACHECK
*
ANALYZE  LR    8,15           SAVE RC
        CH    8,=H'4'
        BE    NOPROF
         LR    7,1           ADDR OF PROF.
         LTR   4,1            ADDR OF PROFILE IN CSA
         BNZ   MOVEPROF
```

```
          LTR   8,8           RC
          BNZ   NOPROF         NOT AUTHD.
          MVC   LISTG+10(10),=CL10'YES'
          B     NOPROF
MOVEPROF  MODESET KEY=ZERO,MODE=SUP
          L     3,0(4)        LENGTH OF PROFILE
          LA    3,0(3)       CLEAR HI BYTE
          C     3,=F'1024'
          BNH   OK
          ABEND 200,DUMP,STEP
OK        LR    5,3           LENGTH
          LA    2,PROFILE     PROFILE AREA HERE
          MVCL  2,4           MOVE FROM CSA
          L     0,PROFILE     SUBPOOL,LENGTH
          FREEMAIN R,LV=(0),A=(7),RELATED=CSA
          MODESET KEY=NZERO,MODE=PROB
          CH    8,=H'4'       RC FROM RACHECK
          BE    NOPROF
          MVC   LISTH+10(8),PROFILE+84 OWNER
          CH    8,=H'0'
          BNE   NOAUTH
          MVC   LISTG+10(10),=CL10'YES'      AUTHORITY OK
NOAUTH    CLC   =C'.RACF.MODEL.PROFILE',PROFILE+7
          BNE   NOTMODEL
          MVC   LISTF+10(10),=CL10'MODEL'  MODEL USED
NOTMODEL  L     3,PROFILE+72               INST. DATA OFFSET
          LTR   3,3
          BZ    NOINST
          LH    4,PROFILE(3)                     INST. DATA LENGTH
          BZ    NOINST
          LA    3,PROFILE+2(3)      ADDR. OF ACTUAL INST. DATA
          EX    4,MVCINST
          CLI   0(3),C' '
          BNE   OPEN          DO USE PROFILE, NOT MODEL
NOINST    MVC   LISTF+10(10),=CL10'MODEL'      USE MODEL
          B     OPEN
NOPROF    MVC   LISTF+10(10),=CL10'NOPROFILE'
OPEN      OPEN  (DCB,(OUTPUT))
          PUT   DCB,LISTA
          PUT   DCB,LISTB
          PUT   DCB,LISTC
          PUT   DCB,LISTD
          PUT   DCB,LISTE
          PUT   DCB,LISTF
          PUT   DCB,LISTG
          PUT   DCB,LISTH
          PUT   DCB,LISTI
          CLOSE (DCB)
RETURN    LA    3,ANS
          IKJRLSA (3)
          L     13,SAVE+4
          RETURN (14,12),RC=0
EXIT      LA    3,ANS
          IKJRLSA (3)
          L     13,SAVE+4
          RETURN (14,12),RC=12
LIST      CAMLST NAME,DSN,,WORK
DSN       DC    44C' '
VOLSER    DC    CL6' '
MVCVOL    MVC   WORK+6(1),0(3)   MOVE VOL PARM
INSTLN    DC    C'COMMAND'  INSTDATA FOR RACHECK- STOPS EXPIRY DATE SIM
```

```
CATP DC C'NOT AUTHORIZED TO SEARCH CATALOG'
         DS    0F
WORK     DC    265C' '
LISTA    DC    CL200'GLOBAL VOL UNIT INST PROF AUTH OWNR VSAM'
LISTB    DC    CL200'CONTROL MSG'
LISTC    DC    CL200'SET &&VOL=ARCHIV'
LISTD    DC    CL200'SET &&UNIT=DISK'
LISTE    DC    CL200'SET &&INST= '
LISTF    DC    CL200'SET &&PROF=PROFILE'
LISTG    DC    CL200'SET &&AUTH=NO'
LISTH    DC    CL200'SET &&OWNR='
LISTI    DC    CL200'SET &&VSAM='
MVCG     MVC   1(1,4),0(2)
*
NODSN DC C'DATASET NAME NOT FOUND IN CATALOG OR ARCHIVE CATALOG'
ARCHPH DC C'PHYSICAL ERROR SEARCHING ARCHIVE CATALOG'
ARCHLO DC C'LOGICAL ERROR SEARCHING ARCHIVE CATALOG'
ARCHOP DC C'ERROR OPENING ARCHIVE CATALOG'
ARCHRC   DS    F
CAT      ACB   DDNAME=ARCHCAT,MACRF=(KEY,DIR)
ARCH     RPL   AREA=RECADDR,AREALEN=4,ARG=DSN,ACB=CAT,         XXXXXXXXXX
               OPTCD=(KEY,DIR,LOC)
RECADDR  DS    F
*
LISTAL   CAMLST NAME,ALIAS,,WORK
LISTOB   CAMLST SEARCH,DSN,VOLOB,WORKOB
VOLOB    DC     CL6' '
WORKOB   DC     CL140' '
VSAMI    DC     X'00'
MSGMNT DC C'DATA SET ON UNMOUNTED VOLUME, COMMAND FAILED'
VTOCIO DC C'PERMANENT I/O ERROR IN VTOC OR INVALID DSCB, FAILED'
NODS DC C'DATASET DOES NOT EXIST, ONLY CATLG ENTRY, FAILED'
ALIAS    DC     CL44' '
VOLUME   DC     CL6' '
SAVE     DC    18F'0'
CPPL     IKJCPPL
CATFIND  CSECT
PROGNAME DC    C'CATFIND '
GFPOINTR DC    A(GFPARMS)
         IKJEFFGF
CATFIND  CSECT
ANS      DC    A(0)
ECB      DC    A(0)
MVCD     MVC   DSN(1),0(2)
PPL      DS    0F
PPLUPT   DS    F
PPLECT   DC    A(0)
PPLECB   DC    A(ECB)
PPLPCL   DC    A(PCL)
PPLANS   DC    A(ANS)
PPLCOM   DS    F
PPLWRK   DS    F
MVCINST  MVC   LISTE+10(1),0(3)
PROFILE  DC    256F'0'
RESOURCE DC    CL44' '
ACCESS   DC    X'00'
AUTHMSG  DC    C'REQUIRED AUTHORITY INVALID'
MSG4     DC    C'CATALOG INACCESSIBLE, UNABLE TO CONTINUE'
MSG20    DC    C'SYNTAX ERROR IN DATASET NAME, UNABLE TO CONTINUE'
MSG24    DC    C'CATALOG ERROR, UNABLE TO CONTINUE'
MSG28    DC    C'UNKNOWN CATALOG ERROR, UNABLE TO CONTINUE'
```

```
DCB      DCB   DDNAME=$@ 99$@ ,DSORG=PS,MACRF=(PM),LRECL=200,     XXXXXXXX
               BLKSIZE=9000,RECFM=FB
         PRINT  NOGEN
PCL      IKJPARM
DSNM     IKJPOSIT DSNAME,USID,PROMPT='DATA SET NAME'
VOL      IKJKEYWD
         IKJNAME  'VOL',SUBFLD=VOLA
AUTH     IKJKEYWD
         IKJNAME  'AUTH',SUBFLD=AUTHA
GEN      IKJKEYWD
         IKJNAME  'GEN',SUBFLD=GENA
VOLA     IKJSUBF
VOLB     IKJIDENT 'VOLSER',MAXLNTH=6,OTHER=ALPHANUM
AUTHA    IKJSUBF
AUTHB    IKJIDENT 'ACCESS AUTHORITY REQUIRED',MAXLNTH=8
GENA     IKJSUBF
GENB     IKJIDENT 'GENERATION NO.',FIRST=ANY,OTHER=ANY
         IKJENDP
         CVT   DSECT=YES
         END
//LKED.SYSLMOD DD DSN=SYS1.WRELINK(CATFIND),DISP=SHR
//LKED.SYSPRINT DD SYSOUT=*
//LKED.SYSLIB DD DSN=SYS1.ARCHIVE.LOAD,DISP=SHR
```

SHARE CLIST

```
PROC 1 DSN OWNER() UACC() ARCHIVE ID() ACCESS() DELETE FROM() +
DEFAULT FROMDEFAULT GDG PROMPT REPEAT GENERATION()
ATTN EXIT
ERROR GOTO END
GLOBAL VVV UUU INST PROF AUTH OWNR VSAM
CONTROL MAIN NOMSG
/*CONTROL LIST CONLIST MSG PROMPT
PROF WTP
IF &UACC= && &ID= && &ACCESS¬= THEN DO
     SET &UACC=&ACCESS
     SET &ACCESS=
     WRITE ID PARM OMITTED, ACCESS PARM CHANGED TO UACC
     END
IF &OWNER&UACC&ACCESS&DELETE&FROM&DEFAULT&FROMDEFAULT= THEN DO
  WRITE NO PARAMETERS WERE INCLUDED TO ALTER THE ACCESS TO THE DATASET - TRY AGAIN.
  EXIT
  END
SET &DEF=&DEFAULT
IF &FROMDEFAULT ¬= | &STR(&FROM)=&STR(*) THEN DO
      IF &SUBSTR(1,&STR(&DSN))=' THEN SET &FROM='&SUBSTR(2:4,&STR(&DSN)).RACF.MODEL.PROFILE
      ELSE SET &FROM=RACF.MODEL.PROFILE
      SET &FCLASS=DATASET
      END
 ELSE IF &FROM ¬= THEN DO
     FILE FI($@ 99$@ ) DA('&SYSUID..$@ 99$@ .CLIST') FXD LRECL(200) NOMSG
     CONTROL MSG
ALLOC F(ARCHCAT) DA('SYSV.ARCHIVE.CATLG') SHR REUSE
     CATFIND &FROM
FREE F(ARCHCAT)
     CONTROL NOMSG
     EX '&SYSUID..$@ 99$@ .CLIST'
     DEL '&SYSUID..$@ 99$@ .CLIST'
     IF &UUU=TAPE THEN DO
          SET &FROM=&VVV
          SET &FCLASS=TAPEVOL
          END
     ELSE SET &FCLASS=DATASET
     END
PRMPT: +
CONTROL MSG
SET &VOL=
IF &ARCHIVE ¬= THEN SET &VOL=ARCHIV
IF &GDG ¬= THEN SET &VOL=DUMMY
IF &SUBSTR(1,&STR(&DSN))=&STR(*) THEN SET &DSN=RACF.MODEL.PROFILE
SET &L=&LENGTH(&STR(&DSN))
IF &L>6 THEN SET &L=6
IF &STR(&DSN)=RACF.MODEL.PROFILE | +
   &SUBSTR(&L:&LENGTH(&STR(&DSN)),&STR(&DSN))=RACF.MODEL.PROFILE'  THEN DO
IF &UACC ¬= THEN WRITE YOUR DEFAULT UACC MAY NOT BE CHANGED FROM 'NONE'
IF &OWNER ¬= THEN ALD &DSN OWNER(&OWNER)
IF &ACCESS ¬= THEN PE &DSN ID(&ID) ACCESS(&ACCESS)
IF &DELETE ¬= THEN PE &DSN ID(&ID) DELETE
IF &FROM ¬= THEN PE &DSN FROM(&FROM) FCLASS(&FCLASS)
GOTO END
END
IF &VOL ¬= THEN GOTO VOLUMEA
FILE FI($@ 99$@ ) DA('&SYSUID..$@ 99$@ .CLIST') FXD LR(200) NOMSG
CONTROL MSG
ALLOC F(ARCHCAT) DA('SYSV.ARCHIVE.CATLG') SHR REUSE
```

```
CATFIND &DSN VOL(&VOL) AUTH(ALTER) GEN(&GENERATION)
FREE F(ARCHCAT)
CONTROL NOMSG
EX '&SYSUID..$@ 99$@ .CLIST'
DEL '&SYSUID..$@ 99$@ .CLIST'
SET &VOL=&VVV
SET &UNIT=&UUU
CONTROL MSG
/*WRITE &VVV &UUU &INST &PROF &AUTH &OWNR &VSAM
IF &VSAM=VSAM && &VOL¬=ARCHIV THEN GOTO VSAMDS
IF &UNIT=TAPE THEN GOTO TAPE
IF &DEF ¬= THEN GOTO DEFLT
IF &PROF=PROFILE THEN GOTO VOLUME
GOTO ADDSD
VOLUMEA: +
ERROR GOTO ADDSD
CONTROL NOMSG
VOLUME: +
IF &UACC ¬= THEN ALD &DSN UACC(&UACC) VOL(&VOL)
IF &OWNER ¬= THEN ALD &DSN OWNER(&OWNER) VOL(&VOL)
IF &ACCESS ¬= THEN PE &DSN ID(&ID) ACCESS(&ACCESS) VOL(&VOL)
IF &DELETE ¬= THEN PE &DSN ID(&ID) DELETE VOL(&VOL)
IF &FROM ¬= THEN PE &DSN VOL(&VOL) FROM(&FROM) FCLASS(&FCLASS)
IF &DEF ¬= THEN DD &DSN NOSET VOL(&VOL)
GOTO END
DEFLT: +
CONTROL MSG
IF &DEF ¬= THEN DD &DSN NOSET VOL(&VOL)
GOTO END
ADDSD: +
ERROR GOTO END
CONTROL MSG
AD &DSN NOSET VOL(&VOL) UNIT(DISK)
GOTO VOLUME
VSAMDS: ERROR
IF &SUBSTR(1,&DSN)=' THEN DO
   SET &DSND=&SUBSTR(2:&LENGTH(&STR(&DSN))-1,&STR(&DSN))
   SET &DSNI='&STR(&DSND).INDEX'
   SET &DSND='&STR(&DSND).DATA'
   END
 ELSE DO
   SET &DSND=&STR(&DSN).DATA
   SET &DSNI=&STR(&DSN).INDEX
   END
IF &DEF¬= THEN GOTO DVSAM
IF &PROF=PROFILE THEN GOTO ALTVSAM
ADVSAM: ERROR
CONTROL MSG
AD &DSN NOSET
AD &DSND N
AD &DSNI N
ALTVSAM: +
IF &UACC ¬= THEN ALD &DSN UACC(&UACC)
IF &OWNER ¬= THEN ALD &DSN OWNER(&OWNER)
IF &ACCESS ¬= THEN PE &DSN ID(&ID) ACCESS(&ACCESS)
IF &DELETE ¬= THEN PE &DSN ID(&ID) DELETE
IF &FROM ¬= THEN PE &DSN FROM(&FROM) FCLASS(&FCLASS)
IF &UACC ¬= THEN ALD &DSND UACC(&UACC)
IF &OWNER ¬= THEN ALD &DSND OWNER(&OWNER)
IF &ACCESS ¬= THEN PE &DSND ID(&ID) ACCESS(&ACCESS)
IF &DELETE ¬= THEN PE &DSND ID(&ID) DELETE
IF &FROM ¬= THEN PE &DSND FROM(&FROM) FCLASS(&FCLASS)
IF &UACC ¬= THEN ALD &DSNI UACC(&UACC)
IF &OWNER ¬= THEN ALD &DSNI OWNER(&OWNER)
IF &ACCESS ¬= THEN PE &DSNI ID(&ID) ACCESS(&ACCESS)
IF &DELETE ¬= THEN PE &DSNI ID(&ID) DELETE
IF &FROM ¬= THEN PE &DSNI FROM(&FROM) FCLASS(&FCLASS)
GOTO END
DVSAM: +
DD &DSN N
DD &DSND N
DD &DSNI N
GOTO END
TAPE: ERROR GOTO END
CONTROL MSG
SET &I=1
SET &VVOL=&VOL
SET &LEN=&LENGTH(&VVOL)
LOOP: +
SET &L=&LEN
IF &I>&L THEN GOTO END
IF &L>&I+5 THEN SET &L=&I+5
SET &VOL=&SUBSTR(&I:&L,&VVOL)
SET &I=&I+6
IF &SUBSTR(1,&STR(&DSN))=' THEN SET &IN=&SUBSTR(2:4,&STR(&DSN))
                     ELSE SET &IN=&SUBSTR(1:3,&SYSPREF)
IF &DEF ¬= THEN GOTO TDEF
IF &UACC&ID&FROM¬= THEN DO
IF &UACC ¬= THEN RALT TAPEVOL (&VOL) UACC(&UACC)
IF &OWNER ¬= THEN RALT TAPEVOL (&VOL) OWNER(&OWNER)
IF &ACCESS ¬= THEN PE &VOL CLASS(TAPEVOL) ID(&ID) ACCESS(&ACCESS)
IF &DELETE ¬= THEN PE &VOL CLASS(TAPEVOL) ID(&ID) DELETE
IF &FROM ¬= THEN PE &VOL CLASS(TAPEVOL) FROM(&FROM) FCLASS(&FCLASS)
RALT TAPEVOL (&VOL) DATA('$&IN     ')
END
ELSE DO
  IF &OWNER¬= THEN RALT TA (&VOL) OWNER(&OWNER)
  IF &SUBSTR(1,&INST)=$ THEN RALT TA (&VOL) DATA('$&IN     ')
                        ELSE RALT TA (&VOL) DATA(' &IN     ')
  END
GOTO LOOP
TDEF: +
RDEL TA (&VOL)
RDEF TAPEVOL (&VOL) DATA(' &IN     ')
GOTO LOOP
END: ERROR EXIT
IF &REPEAT&PROMPT = THEN GOTO EXIT
WRITE ENTER DSN
READ &DSN
IF &STR(&DSN)= THEN GOTO EXIT
GOTO PRMPT
EXIT: WRITE SHARE COMMAND COMPLETE, USE LISTP TO VERIFY.
```

LISTP CLIST

```
PROC 1 DSN ID() PREFIX() AUTHUSER ARCHIVE GDG GENERATION() NAMES
ATTN EXIT
CONTROL MSG MAIN
/*CONTROL LIST CONLIST PROMPT
ERROR EXIT
GLOBAL VOL UNIT INST PROF AU OWNR VSAM
IF &ARCHIVE ¬= THEN SET &ARCHIVE=ARCHIV
IF &GDG ¬= THEN SET &ARCHIVE=DUMMY
IF &ID&PREFIX= THEN SET &UID=&SYSPREF
                ELSE SET &UID=&ID&PREFIX
SET &AUTHUSER=AUTH
IF &NAMES¬= THEN GOTO SEARCH
IF &SUBSTR(1,&STR(&DSN))=&STR(*) THEN GOTO LISTPROF
IF &STR('&DSN')=&STR('(NAMES)') THEN GOTO SEARCH
IF &STR('&DSN')=&STR('(DISK)') THEN GOTO DISK
IF &STR('&DSN')=&STR('(ALL)') THEN GOTO ALL
FILE NOMSG DA('&SYSUID..$@ 99$@ .CLIST') FI($@ 99$@ ) FXD LR(200)
ALLOC F(ARCHCAT) DA('SYSV.ARCHIVE.CATLG') SHR REUSE
CATFIND &DSN VOL(&ARCHIVE) GEN(&GENERATION)
FREE F(ARCHCAT)
EX '&SYSUID..$@ 99$@ .CLIST'
/*WRITE VOL UNIT INST PROF AUTH OWNER VSAM
/*WRITE &VOL &UNIT &INST &PROF &AU &OWNR &VSAM
CONTROL NOMSG
DEL '&SYSUID..$@ 99$@ .CLIST'
CONTROL MSG
IF &AU=NO && &PROF=NOPROFILE THEN GOTO NOMODEL
IF &PROF ¬=PROFILE THEN GOTO MODEL
IF &UNIT=TAPE THEN DO
                   SET &L=&LENGTH(&VOL)
                   IF &L>6 THEN SET &L=6
                   RL TA &SUBSTR(1:&L,&VOL) &AUTHUSER
                   END
              ELSE LD DA(&DSN) &AUTHUSER
EXIT
SEARCH: WRITE
WRITE
WRITE A LIST OF THE DISK DATA SETS SPECIFICALLY DEFINED TO RACF FOR &UID
WRITE
SR MASK(&UID)
EXIT
ALL: +
CONTROL NOMSG
E '&SYSUID..$@ 88$@ .DATA' DA EMODE
10  LISTC LVL(&UID)
END S
FILE NOMSG FI(SYSIN) DA('&SYSUID..$@ 88$@ .DATA')
FILE NOMSG FI(SYSPRINT) DA('&SYSUID..$@ 88$@ .LISTC')
ERROR
CALL 'SYS1.LINKLIB(IDCAMS)'
IF &LASTCC>0 THEN GOTO LISTCERR
FILE NOMSG FI(SYSIN) DA(*)
FILE NOMSG FI(SYSPRINT) DA(*)
FILE NOMSG FI(LISTCATG) DA('&SYSUID..$@ 88$@ .LISTC')
FILE NOMSG DA('&SYSUID..$@ 99$@ .CLIST') FI($@ 99$@ ) FXD LR(200)
ALLOC F(ARCHCAT) DA('SYSV.ARCHIVE.CATLG') SHR REUSE
OPENFILE LISTCATG
REPEAT: +
CONTROL NOMSG
```

```
ERROR GOTO ARCHV
GETFILE LISTCATG
ERROR GOTO REPEAT
IF &SUBSTR(2:8,&STR(&LISTCATG)) ¬= NONVSAM THEN GOTO REPEAT
SET &DSN=&SUBSTR(18:&LENGTH(&STR(&LISTCATG)),&STR(&LISTCATG))
CATFIND '&DSN'
EX '&SYSUID..$@ 99$@ .CLIST'
IF &UNIT¬=TAPE THEN GOTO REPEAT
IF &PROF ¬=PROFILE THEN GOTO REPEAT
CONTROL MSG
WRITE
WRITE DATA SET &DSN
SET &L=&LENGTH(&VOL)
IF &L>6 THEN SET &L=6
RL TA &SUBSTR(1:&L,&VOL) &AUTHUSER
GOTO REPEAT
ARCHV: ERROR EXIT
CLOSFILE LISTCATG
DEL '&SYSUID..$@ 88$@ .LISTC'
DEL '&SYSUID..$@ 99$@ .CLIST'
DEL '&SYSUID..$@ 88$@ .DATA'
FREE F(ARCHCAT)
DISK: +
CONTROL MSG
LD &AUTHUSER PREFIX(&UID)
EXIT
MODEL: +
IF &SUBSTR(1,&STR(&DSN))=' THEN SET &UID=&SUBSTR(2:4,&STR(&DSN))
WRITE
WRITE THE DATASET HAS NOT BEEN SPECIFICALLY PROTECTED USING THE SHARE
WRITE COMMAND AND HAS DEFAULT PROTECTION ATTRIBUTES.
LISTPROF: +
WRITE THE DEFAULT PROTECTION ATTRIBUTES ARE
WRITE DEFINED FOR THE DUMMY DATASET :
WRITE  &UID..RACF.MODEL.PROFILE AND ARE LISTED BELOW :
WRITE (NOTE THAT ACCESS TO SPECIFICALLY DEFINED DATA SETS IS NOT
WRITE  CONTROLLED BY THIS DEFAULT).
WRITE
SET &UID=&SUBSTR(1:3,&UID)
LD DA('&UID..RACF.MODEL.PROFILE') &AUTHUSER
EXIT
LISTCERR: ERROR EXIT
L '&SYSUID..$@ 88$@ .LISTC'
DEL '&SYSUID..$@ 88$@ .LISTC'
DEL '&SYSUID..$@ 88$@ .DATA'
NOMODEL: WRITE DATASET HAS NOT BEEN DEFINED SPECIFICALLY USING THE SHARE
WRITE COMMAND AND THE OWNER DOES NOT HAVE A DEFAULT MODEL DEFINED
WRITE TO RACF - SEE THE DUTY PROGRAMMER.
```

```
         TITLE 'IKJEFF10 - TSO SUBMIT USER EXIT, RACF PASSWORD VERSION'
*         R.J. WHATMOUGH  - LAST CHANGE 18/4/79.
* FUNCTION -
*     THIS MODULE INSPECTS AND MODIFIES JCL CARDS SUBMITTED FOR
*     BACKGROUND PROCESSING USING THE TSO SUBMIT COMMAND.
*     THE JOBNAME IS FORCED TO START WITH THE CURRENT USERID.
*     IF THE OPERAND FIELD OF A JOB CARD IS IN SUITABLE FORM,
*     THE ACCOUNTING AND PROGRAMMER NAME FIELDS ARE INSERTED, AS
*     FOLLOWS.....
*       OLD OPERAND...    NEW OPERAND...
*         BLANK OR ","      "'ACCT',USERID"
*         ",,"              "'ACCT',USERID,"
*         ",,XXX"           "'ACCT',USERID," CONTINUED "// XXX"
*     IF A JOB STATEMENT DOES NOT INCLUDE A 'PASSWORD' PARAMETER,
*     THE USER'S CURRENT PASSWORD IS SUPPLIED ON AN ADDITIONAL
*     CARD AT THE END OF THE STATEMENT. IF THE LAST CARD DOES NOT LEAVE
*     ROOM FOR A COMMA TO BE ADDED, AN ERROR MESSAGE IS ISSUED AND THE
*     JOB IS CANCELLED.
*     IF ANY NOTIFY= OR USER= PARAMETER IS SUPPLIED, IT IS CHANGED TO
*     THE CURRENT USERID.
*
* ENTRY CONDITIONS -
*
*     KEY 1, SUPERVISOR STATE
*     R15 = A(IKJEFF10)
*     R14 = A(RETURN POINT)
*     R13 = A(SAVE AREA)
*     R1  = A(PARAMETER LIST DESCRIBED IN SYSTEM MACRO IKJEFFIE)
*
* EXIT CONDITIONS -
*     R15 = RETURN CODE INDICATE CONTINUE PROCESSING STATEMENT,
*           CONTINUE AND INSERT ANOTHER STATEMENT, ISSUE MESSAGE
*           AND CALL AGAIN, OR TERMINATE SUBMIT.
*     OTHER REGISTERS RESTORED.
*     JOB CARD CONTENTS POSSIBLY CHANGED.
*     CARD IMAGE POINTER IN PARAMETER LIST SET IF CARD INSERTED.
*     EXIT WORK FIELD OF PARAMETER LIST IN USE.
*     WORKING STORAGE GOTTEN OR FREED (SUBPOOL 0).
*
* ATTRIBUTES -
*     RE-ENTERABLE, RE-USEABLE, REFRESHABLE
*
* EXTERNAL REFERENCES -
*     EXIT PARAMETER LIST
*     JCL CONTROL INFORMATION
*
* REGISTER USAGE -
*
*     R2   - A(USERID)
*     R3-R8 - WORK REGISTERS
*     R9   - BASE FOR THIS ROUTINE
*     R10 - CONTROL BYTE BASE
*     R11 - CARD IMAGE ADDRESS
*     R12 - PARAMETER LIST BASE
*     R13 - SAVE AREA
*     R14 - RETURN ADDRESS
*
* METHOD -
*
*   SAVE REGISTERS
*   IF CANCEL NOT REQUIRED THEN
*     FIND JCL CARD IMAGE.
*     IF A(IMAGE) NON-ZERO THEN
*       IF JOB CARD THEN
*         IF NOT A CONTINUATION THEN
*           INDICATE PASSWORD FOUND, NOT TO BE ADDED.
*           FIND ACEE FOR USER (IF ANY).
*           IF USER DEFINED TO RACF,
*             FIND ACEEIEP.
*             IF PASSWORD STORED (PASSWORD SYSTEM OPERATING),
*               INDICATE PASSWORD NOT FOUND.
*             ENDIF
*           ENDIF
*           COPY USERID TO COLS 3-5.
*           IF OPERAND FIELD PRESENT AND
*           START COL <= 69 AND
*           FIELD IS COMMA-COMMA-NONBLANK THEN INSERT REQUIRED,
*             IF NO STORAGE GOTTEN THEN
*               INDICATE STORAGE GOTTEN.
*               GET STORAGE FOR INSERTS.
*               INDICATE NULL CARDS TO BE PROCESSED.
*             ENDIF
*             SET INSERT TO SLASH-SLASH-BLANKS.
*             COPY STATEMENT (OPERAND COLUMN+2 TO COL 71)
*             TO INSERT (BEGINNING COLUMN 4).
*             INDICATE INSERT REQUIRED.
*           ENDIF
*           IF OPERAND FIELD PRESENT AND
*           START COLUMN <=70 AND
*           OPERAND BEGINS COMMA-COMMA THEN
*             SET MARK TO COMMA.
*           ELSE
*             SET MARK TO BLANK.
*           ENDIF
*           IF OPERAND FIELD NOT PRESENT ORIF
*           START COLUMN = 71 AND
*           CHARACTER IS A COMMA ORIF
*           START COLUMN <= 70 AND
*           FIRST CHARACTER IS A COMMA AND
*           SECOND CHARACTER IS COMMA OR BLANK THEN
*             FIND FIRST BLANK ON CARD (OR FORCE ONE AT COLUMN 11).
*             INSERT 'JOB' AFTER BLANK.
*             CLEAR AFTER 'B' TO COLUMN 72.
*             SET OPERAND START COL. = 2 AFTER 'B'.
*             INSERT ACCOUNTING INFORMATION, COMMA AND USERID 2 COLUMNS
*             AFTER 'B'.
*             INSERT MARK AFTER USERID.
*           ENDIF
*         ENDIF (NO CHANGE TO CONTINUATION OF ORIGINAL JOB CARD)
*       ELSE NULL CARD
*         FREE STORAGE FOR INSERTS.
*         INDICATE NO STORAGE GOTTEN.
*         INDICATE NULL CARDS NOT TO BE PROCESSED.
*       ENDIF
*     ELSE INSERT TO BE PASSED NOW
*       IF PASSWORD TO BE ADDED,
*         SET INSERT TO '// PASSWORD=',BLANKS.
*         FIND ACEE.
*         FIND ACEEIEP.
*         ADD PASSWORD TO INSERT.
*         INDICATE PASSWORD FOUND, NOT TO BE ADDED.
*       ELSE PARAMETERS FROM FIRST CARD YET TO BE SCANNED,
*         SET OPERAND COLUMN = 4.
*       ENDIF
*       PUT A(INSERT) IN PARAMETER LIST.
*       INDICATE NO INSERT REQUIRED.
*     ENDIF
*     IF A JOB CARD AND NOT INTERNAL COMMENT THEN
*       SET CURRENT COLUMN = OPERAND START COLUMN.
*       INDICATE SCAN NOT DONE, NOT QUOTED STRING.
*       DO UNTIL SCAN DONE,
*         IF CURRENT COLUMN LESS THAN 72,
*           SEARCH FROM CURRENT COL. TO 71 FOR QUOTE, BLANK OR '='.
*         ELSE
*           ASSUME NOTHING FOUND.
*         ENDIF
*         IF CHARACTER FOUND,
*           SET CURRENT COLUMN = FOUND COLUMN + 1.
*           IF QUOTE FOUND THEN
*             REVERSE QUOTED STRING INDICATOR.
*           ELSE
*             IF NOT QUOTED STRING THEN
*               IF '=' THEN
*                 IF COLUMN 12 OR LATER
*                 AND PREVIOUS 8 COLUMNS ARE 'PASSWORD' THEN
*                   INDICATE PASSWORD FOUND.
*                 ELSE NOT PASSWORD,
*                   IF COLUMN 8 OR LATER
*                   ANDIF PREVIOUS 4 COLUMNS ARE 'USER'
*                   OR PREVIOUS 6 COLUMNS ARE 'NOTIFY',
*                     COPY USERID TO NEXT 3 COLUMNS.
*                   ENDIF
*                 ENDIF
*               ELSE BLANK FOUND,
*                 SET CURRENT COL. = FOUND COL.
*                 INDICATE SCAN DONE.
*               ENDIF
*             ENDIF
*           ENDIF
*         ELSE NO SPECIAL CHARACTER FOUND,
*           INDICATE SCAN DONE.
*           SET CURRENT COLUMN = 72.
*         ENDIF
*       ENDDO (CURRENT COL. IS LAST COL. OF OPERAND + 1)
*       IF PASSWORD NOT FOUND
*       AND NO INSERT REQUIRED ALREADY
*       AND LAST OPERAND COLUMN WAS NOT A COMMA THEN
*         IF CURRENT COLUMN IS 72 THEN
*           IF STORAGE GOTTEN FOR INSERTS THEN
*             FREE STORAGE.
*           ENDIF
*           INDICATE CANCEL REQUIRED NEXT ENTRY.
*           PUT A(NO-ROOM MESSAGE) IN PARMLIST.
*           RESTORE REGISTERS.
*           RETURN INDICATING MESSAGE TO BE ISSUED.
*         ENDIF
*         FORCE CURRENT COLUMN AND NEXT = ', '
*         INDICATE PASSWORD TO BE ADDED.
*         IF NO STORAGE GOTTEN THEN
*           INDICATE STORAGE GOTTEN.
*           GET STORAGE FOR INSERTS.
*           INDICATE NULL CARDS TO BE PROCESSED.
*         ENDIF
*         INDICATE INSERT REQUIRED.
*       ENDIF
*     ENDIF
*     IF INSERT REQUIRED THEN
*       RESTORE REGISTERS
*       RETURN INDICATING INSERT.
*     ELSE
*       RESTORE REGISTERS.
*       RETURN INDICATING CONTINUE PROCESSING.
*     ENDIF
*   ELSE CANCEL REQUIRED, MESSAGE HAS BEEN ISSUED.
*     RESTORE REGISTERS.
*     RETURN INDICATING CANCEL.
*   ENDIF
*
* NOTES -
*   1)THE COMMAND PROCESSOR GENERATES A NULL CARD AT THE END OF
*     THE LAST JOB. THIS ROUTINE USES NULL CARDS AS AN OPPORTUNITY
*     TO FREE GOTTEN STORAGE.
*   2)THE RECONSTRUCTED ACCOUNTING FIELD INCLUDES THE QUOTES.
*   3)JOB CARD COLUMN NUMBERS START AT 1.
*   4)THE CURRENT USERID IS TAKEN FROM THE CURRENT ASCB, AND NOT
*     FROM THE IEUSRIDP FIELD OF THE PARAMETER LIST, TO ALLOW
*     SUBMIT TO BE ISSUED UNDER THE TSO COMMAND PACKAGE.
         EJECT
IKJEFF10 CSECT
         PRINT NOGEN
         SAVE  (14,12),,*               SAVE REGISTERS.
         LR    R9,R15                   LOAD BASE REGISTER.
         USING IKJEFF10,R9
         L     R12,0(R1)                FIND PARAMETERS.
         USING IEEXITL,R12              PARAMETER BASE.
         L     R10,IESUBCTP             FIND CONTROL BYTES.
         USING IESUBCTD,R10             CONTROL BYTE BASE.
         TM    IEEXITWD,CANCEL          IF NOT CANCEL AFTER MESSAGE,
         BO    A460
         L     R11,IECARDP              FIND CARD IMAGE
         LTR   R11,R11                  IF PRESENT,
         BZ    A130
         BCTR  R11,0                    OFFSET FOR COLUMN NUMBERING.
         TM    IESTMTYP,IESJOB          IF JOB CARD,
         BZ    A110
         TM    IESTMTYP,IESCONTN        IF NOT CONTINUATION,
         BO    A100
*----------------------------------------------------------------------
*     PROCESS FIRST LINE OF JOB STATEMENT
*----------------------------------------------------------------------
         L     R2,16                    FIND CVT.
         L     R2,0(R2)                 FIND TCB-ASCB LIST (CVTTCBP).
         L     R3,12(R2)                FIND CURRENT ASCB.
*                                       ASSUME 3-BYTE USERID IN ASCB
*                                       JOBNAME.
*----------------------------------------------------------------------
*     CHECK WHETHER A PASSWORD CAN BE SUPPLIED
*----------------------------------------------------------------------
         OI    IEEXITWD,PWFND           INDICATE PASSWORD FOUND.
         NI    IEEXITWD,ALL-PWADD       INDICATE DON'T ADD ONE.
         L     R4,108(R3)               FIND ASXB.
         L     R4,200(R4)               FIND ACEE.
         LTR   R4,R4                    IF USER DEFINED TO RACF,
         BZ    A004
         L     R4,12(R4)               FIND ACEEIEP.
         LA    R4,0(R4)
         LTR   R4,R4                   IF PASSWORD STORED,
         BZ    A003
         NI    IEEXITWD,ALL-PWFND      INDICATE PASSWORD NOT FOUND.
A003     EQU   *                       ENDIF
A004     EQU   *                       ENDIF
         EJECT
*----------------------------------------------------------------------
*     CHECK JOB NAME
*----------------------------------------------------------------------
         L     R2,172(R3)              FIND BATCH JOBNAME (ASCBJBNI)
         LTR   R2,R2                   IF NONE,
         BNZ   A005
         L     R2,176(R3)              FIND LOGON JOBNAME (ASCBJBNS)
A005     EQU   *
         MVC   3(3,R11),0(R2)          FORCE USERID INTO JOBNAME.
*----------------------------------------------------------------------
*     MOVE EXISTING PARMS TO AN INSERT LINE
*----------------------------------------------------------------------
         SR    R3,R3                   GET FIRST OPERAND COLUMN NO.
         IC    R3,IEOPRAND
         LA    R4,0(R11,R3)            FIND OPERAND IF ANY.
         LTR   R3,R3                   IF OPERAND PRESENT,
         BZ    A040
         C     R3,=F'70'               AND NOT AFTER COL. 69,
         BNL   A040
         CLC   0(2,R4),=C',,'          AND COMMA-COMMA,
         BNE   A040
         CLI   2(R4),C' '              AND NOT BLANK FOLLOWING,
         BE    A040
         TM    IEEXITWD,GOTTEN         INSERT REQUIRED.
         BO    A030                    IF STORAGE NOT GOTTEN,
         OI    IEEXITWD,GOTTEN         INDICATE GOTTEN NOW.
         GETMAIN R,LV=80               GET INSERT STORAGE.
         STCM  R1,7,IEEXITWD+1         PUT ADDRESS IN USER WORD.
         OI    IETAKEEX,IETNULL        INDICATE PASS NULL CARDS.
A030     EQU   *                       ENDIF.
         L     R1,IEEXITWD             FIND INSERT STORAGE.
         MVC   0(3,R1),=C'// '         SET INSERT TO NULL.
         MVC   3(77,R1),2(R1)
         LA    R5,2(R4)                FIND OPERAND COLUMN 3.
         LA    R6,71(R11)              FIND COL 71 OF CARD.
         SR    R6,R5                   GET LENGTH-1.
         EX    R6,MVINSRT              PUT REST OF OPERAND IN INSERT.
         OI    IEEXITWD,INSERT         INDICATE INSERT REQUIRED.
A040     EQU   *                       ENDIF. JOB CARD CAN BE CHANGED.
         EJECT
*----------------------------------------------------------------------
*     ADD ACCOUNTING PARAMETERS
*----------------------------------------------------------------------
         LTR   R3,R3                   IF OPERAND PRESENT AND
         BZ    A042
         C     R3,=F'70'               START COLUMN <= 70 AND
         BH    A042
         CLC   0(2,R4),=C',,'          AND COMMA-COMMA THEN
         BNE   A042
         LA    R8,C','                 MAKE MARK A COMMA.
         B     A044
A042     EQU   *                       ELSE
         LA    R8,C' '                 MAKE MARK A BLANK.
```

```
A044     EQU   *                       ENDIF.
         LTR   R3,R3                   IF OPERAND NOT PRESENT
         BE    A050                    OR,
         C     R3,=F'70'               IF OPERAND IN COL. 71,
         BNH   A045
         CLI   0(R4),C','              AND A COMMA
         BE    A050                    OR
         B     A090
A045     EQU   *                       NOT AFTER COL. 70,
         CLI   0(R4),C','              AND FIRST CHAR IS COMMA,
         BNE   A090
         CLI   1(R4),C','              AND SECOND IS COMMA OR BLANK,
         BE    A050
         CLI   1(R4),C' '
         BNE   A090
A050     EQU   *                       THEN,
         LA    R5,3(R11)               FIND COLUMN 3.
         LA    R6,11(R11)              FIND COLUMN 11.
A060     EQU   *                       FOR EACH COLUMN,
         CLI   0(R5),C' '              TEST FOR BLANK,
         BE    A070                    UNTIL ONE FOUND,
         LA    R5,1(R5)                OR AT COLUMN 11,
         CR    R5,R6
         BL    A060
         MVI   0(R5),C' '              IN WHICH CASE FORCE ONE.
A070     EQU   *
         MVC   1(4,R5),=C'JOB '        PUT IN OPERATION.
         LA    R6,5(R5)                FIND NEW OPERAND START.
         LA    R7,72(R11)              FIND COL. 72 OF CARD.
         SR    R7,R6                   GET LENGTH TO CLEAR, -1.
         EX    R7,CLRCARD              CLEAR REST OF CARD.
         LR    R7,R6                   FIND OPERAND START COLUMN.
         SR    R7,R11
         STC   R7,IEOPRAND             UPDATE VALUE SUPPLIED.
         L     R4,IEACCTIP             FIND ACCOUNTING INFO.
         L     R5,IEACCTLP             GET LENGTH OF INFO.
         LH    R5,0(R5)
         BCTR  R5,0                    GET LENGTH-1.
         EX    R5,MVACCT               PUT ACCT. INFO. IN OPERAND.
         LA    R6,1(R6,R5)             FIND NEXT COLUMN.
         MVI   0(R6),C','              ADD COMMA.
         MVC   1(3,R6),0(R2)           ADD USERID (3 CHARS).
         STC   R8,4(R6)                ADD MARK, BLANK OR COMMA.
A090     EQU   *                       ENDIF. JOB CARD NOW READY.
A100     EQU   *                       ENDIF. NO CHANGE TO CONTN. CARD.
         B     A120
A110     EQU   *                       ELSE MUST BE NULL CARD.
         EJECT
*---------------------------------------------------------------
*     PROCESS NULL CARD
*---------------------------------------------------------------
         L     R3,IEEXITWD             FIND INSERT STORAGE.
         LA    R3,0(R3)                INDICATE SUBPOOL 0.
         FREEMAIN R,LV=80,A=(3)        FREE STORAGE.
         NI    IEEXITWD,ALL-GOTTEN     INDICATE NO STORAGE.
         NI    IETAKEEX,ALL-IETNULL    INDICATE DON'T PASS NULL CARDS.
A120     EQU   *                       ENDIF.
         B     A440
A130     EQU   *                       ELSE, INSERT NOW REQUIRED.
*---------------------------------------------------------------
*     INSERT A LINE
```

```
*---------------------------------------------------------------------
         L     R1,IEEXITWD             FIND INSERT.
         TM    IEEXITWD,PWADD          IF INSERT WILL BE PASSWORD,
         BZ    A132
         MVC   0(13,R1),=C'// PASSWORD= '   SET UP KEYWORD.
         MVC   13(67,R1),12(R1)        CLEAR REST OF CARD.
         L     R4,16                   FIND CVT.
         L     R4,0(R4)                FIND TCB-ASCB LIST.
         L     R4,12(R4)               FIND ASCB.
         L     R4,108(R4)              FIND ASXB.
         L     R4,200(R4)              FIND ACEE.
         L     R4,12(R4)               FIND ACEEIEP.
         SR    R5,R5                   GET PASSWORD LENGTH - 1.
         IC    R5,8(R4)
         BCTR  R5,0
         EX    R5,MVPSWD               PUT PASSWORD AFTER '='
         OI    IEEXITWD,PWFND          INDICATE PASSWORD FOUND.
         NI    IEEXITWD,ALL-PWADD      INDICATE DON'T ADD PASSWORD.
         B     A134
A132     EQU   *                       ELSE ALLOW SCAN OF MOVED PARMS,
         MVI   IEOPRAND,4              SET OPERAND START TO COL. 4.
A134     EQU   *                       ENDIF.
         LA    R1,0(R1)
         ST    R1,IECARDP              MAKE INSERT THE CARD IMAGE.
         NI    IEEXITWD,ALL-INSERT     INDICATE NO INSERT REQUIRED.
A440     EQU   *                       ENDIF. CARD IMAGE READY.
         EJECT
*---------------------------------------------------------------------
*     IF JOB STATEMENT, LOOK FOR 'PASSWORD=', 'USER=' OR 'NOTIFY='
*---------------------------------------------------------------------
         TM    IESTMTYP,IESJOB         IF JOB STATEMENT,
         BZ    A310
         TM    IESTMTP2,IESCOMNT       AND NOT INTERNAL COMMENT,
         BO    A310
         SR    R1,R1
         SR    R2,R2                   CLEAR R2 FOR TRT.
         L     R4,IECARDP              FIND CARD IMAGE TO BE SCANNED.
         BCTR  R4,0                    OFFSET FOR COLUMN NUMBERING.
         LA    R5,71(R4)               FIND COLUMN 71.
         LR    R7,R4                   FIND COLUMN 0.
         IC    R1,IEOPRAND             SET CURRENT COL. = OPERAND START
         AR    R4,R1
         LA    R0,1                    INDICATE SCAN NOT DONE,
*                                      NOT IN QUOTED STRING.
A150     EQU   *                       DO UNTIL SCAN DONE (R0 = 0),
         LR    R6,R5                   COUNT COLUMNS, CURRENT TO 71.
         SR    R6,R4
         BM    A160                    IF NOT PAST COL. 71,
         SR    R1,R1                   CLEAR R1 FOR TRT.
         EX    R6,TRTJOB               SEARCH FOR SPECIAL CHARS.
         B     A170
A160     EQU   *                       ELSE,
         SR    R1,R1                   SET COND. CODE FOR NOT FOUND.
A170     EQU   *                       ENDIF
         BZ    A240                    IF ANY CHAR. FOUND,
         LA    R4,1(R1)                LET NEXT CHAR. BE THE CURRENT.
         CH    R2,=H'2'                IF A QUOTE,
         BNE   A180
         LCR   R0,R0                   REVERSE QUOTED STRING INDICATION
         B     A230
A180     EQU   *                       ELSE NOT QUOTE,
         LTR   R0,R0                    IF NOT IN QUOTED STRING,
         BM    A220
         CH    R2,=H'2'                 IF '=',
         BL    A200
         SH    R1,=H'8'                 FIND '=' COLUMN - 8.
         LR    R8,R1                    GET NUMBER OF THAT COLUMN.
         SR    R8,R7
         CH    R8,=H'4'                 IF >= 4,
         BL    A182
         CLC   =C'PASSWORD',0(R1)       AND 'PASSWORD' STARTS HERE,
         BNE   A182
         OI    IEEXITWD,PWFND           INDICATE PASSOWRD FOUND.
         B     A190
A182     EQU   *                        ELSE NOT PASSWORD,
         LTR   R8,R8                    IF COLUMN NUMBER >= 0,
         BL    A188
         CLC   =C'USER',4(R1)           ANDIF 'USER' PRECEDED '='
         BE    A184
         CLC   =C'NOTIFY',2(R1)         OR 'NOTIFY' PRECEDED '=',
         BNE   A188
A184     EQU   *
         L     R1,16                    FIND CVT.
         L     R1,0(R1)                 FIND TCB-ASCB LIST.
         L     R1,12(R1)                FIND CURRENT ASCB.
         L     R8,172(R1)               FIND BATCH JOBNAME (ASCBJBNI).
         LTR   R8,R8                    IF NONE,
         BNE   A186
         L     R8,176(R1)               FIND LOGON JOBNAME (ASCBJBNS).
A186     EQU   *
         MVC   0(3,R4),0(R8)            PUT USERID AFTER '='.
A188     EQU   *                        ENDIF
A190     EQU   *                        ENDIF.
         B     A210
A200     EQU   *                        ELSE BLANK FOUND,
         LR    R4,R1                    MAKE IT CURRENT CHAR.
         SR    R0,R0                    INDICATE SCAN DONE.
A210     EQU   *                        ENDIF.
A220     EQU   *                        ENDIF.
A230     EQU   *                        ENDIF, SPECIAL CHAR. PROCESSED.
         B     A250
A240     EQU   *                        ELSE NO CHAR. FOUND.
         SR    R0,R0                    INDICATE SCAN DONE.
         LA    R4,1(R5)                 MAKE CURRENT COL. 72.
A250     EQU   *                        ENDIF
         LTR   R0,R0                    TEST FOR SCAN DONE.
         BNZ   A150                     ENDDO
         EJECT
*---------------------------------------------------------------------
*     IF LAST LINE AND NO PASSWORD, ADD A COMMA.
*---------------------------------------------------------------------
         TM    IEEXITWD,PWFND+INSERT    IF PASSWORD NOT FOUND,
         BNZ   A300                     AND NO INSERT TO COME,
         LR    R6,R4                    AND LAST OP COL. WAS NOT COMMA,
         BCTR  R6,0
         CLI   0(R6),C','
         BE    A300
         CR    R4,R5                    IF CURRENT COLUMN IS 72,
         BNH   A280
         TM    IEEXITWD,GOTTEN          IF INSERT STORAGE TO FREE,
         BZ    A270
         L     R3,IEEXITWD              FIND STORAGE.
         LA    R3,0(R3)                  INDICATE SUBPOOL ZERO.
         FREEMAIN R,LV=80,A=(3)          FREE STORAGE.
A270     EQU   *                         ENDIF.
         OI    IEEXITWD,CANCEL           INDICATE CANCEL.
         LA    R3,PWMESS                 FIND MESSAGE.
         ST    R3,IEMSGP                 PUT ADDRESS IN PARMLIST.
         RETURN (14,12),T,RC=IEMSG       RESTORE AND RETURN WITH MESSAGE.
A280     EQU   *                         ENDIF, ROOM FOR COMMA.
         MVC   0(2,R4),=C', '            PUT COMMA IN CURRENT COL,
*                                        FORCE A BLANK.
         OI    IEEXITWD,PWADD            INDICATE PASSWORD TO BE ADDED.
         TM    IEEXITWD,GOTTEN           IF INSERT STORAGE NOT GOTTEN,
         BO    A290
         OI    IEEXITWD,GOTTEN           INDICATE GOTTEN NOW.
         GETMAIN R,LV=80                 GET INSERT STORAGE.
         STCM  R1,7,IEEXITWD+1           PUT ADDRESS IN USER WORD.
         OI    IETAKEEX,IETNULL          INDICATE PASS NULL CARDS.
A290     EQU   *                         ENDIF, HAVE STORAGE FOR INSERT.
         OI    IEEXITWD,INSERT           INDICATE INSERT REQUIRED.
A300     EQU   *                         ENDIF
A310     EQU   *                         ENDIF, JOB STATEMENT SCANNED.
         EJECT
*----------------------------------------------------------------------
*      RETURN STATEMENT TO OS.
*----------------------------------------------------------------------
         TM    IEEXITWD,INSERT           IF INSERT REQUIRED,
         BZ    A450
         RETURN (14,12),T,RC=IERETURN    RESTORE, RETURN, INSERT IS NEXT.
A450     EQU   *                         ELSE NO INSERT,
         RETURN (14,12),T,RC=IECONTIN    RESTORE, RETURN, USE THIS CARD.
*                                        ENDIF.
A460     EQU   *                         ELSE MESSAGE WAS SENT,
*----------------------------------------------------------------------
*      TELL OS TO CANCEL JOB.
*----------------------------------------------------------------------
         RETURN (14,12),T,RC=IEABORT     RESTORE, RETURN FOR CANCEL.
*                                        ENDIF.
         EJECT
*
* REGISTER EQUATES
*
R0       EQU   0
R1       EQU   1
R2       EQU   2
R3       EQU   3
R4       EQU   4
R5       EQU   5
R6       EQU   6
R7       EQU   7
R8       EQU   8
R9       EQU   9
R10      EQU   10
R11      EQU   11
R12      EQU   12
R13      EQU   13
R14      EQU   14
R15      EQU   15
*
* EQUATES FOR EXIT WORK AREA BYTE 0
*
CANCEL   EQU   X'80'                     CANCEL SUBMIT ON NEXT ENTRY.
```

```
GOTTEN   EQU   X'40'                        STORAGE GOTTEN FOR INSERTS.
INSERT   EQU   X'20'                        INSERT CARD REQUIRED AFTER THIS.
PWFND    EQU   X'10'                        PASSWORD FOUND OR NOT SOUGHT.
PWADD    EQU   X'08'                        INSERTED CARD WILL GIVE PASSWORD
ALL      EQU   X'FF'                        ALL BITS.
*
* INSTRUCTIONS TO BE EXECUTED
*
MVINSRT  MVC   3(0,R1),0(R5)                MOVE OPERAND TO INSERT (COL 4).
CLRCARD  MVC   0(0,R6),4(R5)                CLEAR NEW OPERAND FIELD.
MVACCT   MVC   0(0,R6),0(R4)                PUT ACCT. INFO. IN OPERAND.
MVPSWD   MVC   12(0,R1),9(R4)               PUT PASSWORD IN INSERT COL 13.
TRTJOB   TRT   0(0,R4),TABLE                SCAN JOB CARD FOR SPECIAL CHARS.
*
* CONSTANTS
*
PWMESS   DS    0H                           CAN'T-ADD-PASSWORD MESSAGE
         DC    AL2(EPWMESS-*),C'JOB NOT SUBMITTED - PASSWORD CANNOT '
         DC    C'BE ADDED BECAUSE LAST LINE OF JOB STATEMENT ENDS '
         DC    C'IN COL. 71'
EPWMESS  EQU   *
TABLE    DC    256X'00'                     TRT TABLE, SPECIAL CHAR. SEARCH
         ORG   TABLE+C' '                   BLANK GIVES 1
         DC    X'01'
         ORG   TABLE+C''''                  QUOTE GIVES 2
         DC    X'02'
         ORG   TABLE+C'='                   EQUAL GIVES 3
         DC    X'03'
         ORG   TABLE+256                    END OF TRT TABLE.
         EJECT
*
* DSECTS FOR PARAMETERS
*
         PRINT NOGEN
         IKJEFFIE IETYPE=SUBMIT
         END
```

```
CONTROL MSG MAIN PROMPT
PROF WTP
/**********************************************************************/
/*   THIS PROCEDURE WILL READ A FORMATTED LISTING OF A TSO UADS         */
/*   DATASET AND PRODUCE A DATASET CONTAINING RACF ADDUSER COMMANDS    */
/*   FOR EACH TSO USER WITH HIS EXISTING PASSWORD                      */
/*   USERS WITH NO PASSWORD ARE GIVEN THEIR USER ID AS RACF PASSWORD   */
/*   ----> USE EXECUADS TO EXECUTE THIS CLIST                          */
/**********************************************************************/
SET &F=0                               /* INIT DATA SWITCH */
ERROR DO                               /*SET UP ERROR HANDLING FOR EOF*/
IF &LASTCC=400 THEN GOTO THRU          /*CODE FOR END OF FILE*/
               ELSE DO                 /*ALL OTHERS QUIT WITH MSG*/
               WRITE CLIST FAILED ERROR CODE &LASTCC
               EXIT
               END
      END
ATTN DO
  WRITE CLIST ATTN EXIT
  GOTO THRU
  END
ALLOC DA('UAD.UADS.DATA')    F(IN) SHR       /* PREVIOSLY PRODUCED LISTING*/
ALLOC DA(ALTUSER.CLIST) F(OUT) NEW   /* NEW CONTROL DATASET*/
ALLOC DA(CHGUSER.CNTL)  F(OUTC) NEW  /* NEW CHANGE DATASET*/
OPENFILE IN                            /* OPEN INPUT AND OUTPUT FILES*/
OPENFILE OUT OUTPUT
OPENFILE OUTC OUTPUT
READ:GETFILE IN                        /* READ FIRST RECORD*/
IF &F=1 THEN GOTO OK                   /* TEST START OF DATA SWITCH*/
IF &SUBSTR(2:6,&IN)=&STR(L (*)) THEN  SET &F=1   /*START OF DATA ????*/
          GOTO READ
OK:IF &LENGTH(&IN)<20 THEN GOTO READ   /* CHECK IF RECORD LONG ENOUGH */
IF &SUBSTR(5:6,&IN)=&STR( ) THEN GOTO READ /* CHECK FOR UID IN REC*/
                          ELSE GOTO UID1
UID1:SET &CT=6                        /* SET UP USER ID*/
UID2:IF &SUBSTR(&CT:&CT,&IN)= &STR( ) THEN GOTO GOTUID
SET &CT=&CT+1
GOTO UID2
GOTUID:SET &UID=(&SUBSTR(4:&CT-1,&IN))
LOOP:GETFILE IN                        /* GET NEXT RECORD*/
IF &LENGTH(&IN)<7 THEN GOTO LOOP       /* LONG ENOUGH ???????*/
IF &SUBSTR(7,&IN)=&STR( ) THEN GOTO LOOP /* PASSWORD RECORD ???????*/
PASS1:SET &CT=8                        /* SET UP PASSWORD
PASS2:IF &SUBSTR(&CT:&CT,&IN)= &STR( ) THEN GOTO GOTIT
SET &CT=&CT+1
GOTO PASS2
GOTIT:SET &PASS=(&SUBSTR(6:&CT-1,&IN))
IF &SUBSTR(1,&PASS)=&STR(( THEN SET &PASS=&UID  /* NO PASS SET UID    */
SET &OUT=&STR( ALTUSER )&UID&STR( ADSP CLAUTH(TAPEVOL) )
PUTFILE OUT
SET &OUTC=&STR(//&UID&STR(X) JOB &UID,'228753/135',
// PASSWORD=(DUMMY,&PASS),USER=&UID)
PUTFILE OUTC
SET &OUTC=&STR(//    EXEC BATCHTSO,USERID=JCG,PARM.BATCHTSO=
PUTFILE OUTC
SET &OUTC=&STR(  PROF WTP)
PUTFILE OUTC
GOTO READ                              /* GET NEXT USER ID RECORD*/
THRU:CLOSFILE IN                       /* ALL DONE ... CLEAN UP*/
CLOSFILE OUT                           /* AND GET OUT .....*/
CLOSFILE OUTC
```

APPENDIX VI

INSTALLATION OF THE MODIFICATIONS TO RACF

The full implications should be understood if any of the following instructions are not carried out as defined.

(1) Install RACF according to IBM documentation.

(2) Identify and create all necessary RACF groups. These will include one or more groups for system data sets (in particular group SYS for all SYS1, SYS2 etc. data sets) as well as those groups required for users.

(3) Define all users in the UADS data set to RACF using the CLISTs supplied in Appendix V. Create RACF user definitions for any other users not defined in UADS. Batch jobs submitted from TSO will include the USER parameter on any generated job cards and will fail if the users are not defined to RACF. The ADDUSER command below is suitable for adding users:-

```
ADDUSER userid PASSWORD(password) DFLTGRP(group-name)
CLAUTH(TAPEVOL) GRPACC
```

However note that this command will set the password expired, and it will have to be changed the next time the user accesses the system. If this is considered acceptable then the users will have to be warned that it is going to happen, and instructed on how to change the password. At DRCS this was circumvented by initially setting each user's password to a dummy value and generating and running a batch job for each user that changed the dummy password to his current password in UADS. The jobs consisted simply of a job card with the USER and PASSWORD parameters (the latter nominating the dummy and current passwords) and an EXEC statement to execute IEFBR14.

(4) Modify the RACF exits as required. For example, the exits assume 3 character userids and groupids and contain code to control unusual users at DRCS. In addition tape volume protection is defined only for a range of volume serial numbers.

(5) Install the RACF exits, the SUBMIT exit and the RACF CLISTs and CATFIND command.

(6) Install the RACDEF modification if archive functions will be used with RACF in the same manner as at DRCS.

(7) Install the OPEN modification if tape access control will be used.

(8) Install the SCRIBBLE modifications if privacy control for disk data sets is critical.

(9) Define default profiles for all users and groups, e.g.

```
ADDSD 'userid.RACF.MODEL.PROFILE' NOSET
VOLUME(DUMMY) UNIT(DISK) UACC(ALTER)
```

(The SEARCH command can be used to generate the commands).

(10) Define RACF options including tape volume protection, e.g.

```
SETROPTS CLASSACT(*) TERMINAL(READ) INTERVAL(90)
   NOSTATISTICS(*) INITSTATS AUDIT(*) SAUDIT
      CMDVIOL LIST
```

(11) Create profiles for all existing tape volumes. A suitable command to define a tape profile is:-

```
RDEFINE TAPEVOL(volser) OWNER(ownerid)
      DATA(' userid     ')
```

The DATA parameter must include the userid or groupid of the first data set on the volume with one blank on the left and padded with blanks on the right to a total of 9 characters. The OWNER parameter is the same as the userid if it is a user data set, or identifies the group administrator if it is a group data set. The owner is the only user who can issue the first SHARE command to specifically protect any data set on the volume.
The RDEFINE commands can be automatically created by a program or CLIST that reads and interprets information from the catalogs.

(12) Protect VSAM catalogs and CVOLs e.g.

```
ADDSD 'SYS1.CATALOGA' UACC(UPDATE)
```

The VSAM catalog names must be prefixed by a valid RACF userid or groupid to do this, or the RACF exits must be changed to bypass the naming conventions. It is possible to rename a VSAM catalog by appropriate internal modifications.

(13) Test RACF for selected users by turning on the DSCB protect flag for their DISK data sets e.g.

```
ADDSD dsn
DELDSD dsn NOSET
```

A program or CLIST to automatically generate these commands from catalog or VTOC information greatly reduces the effort involved.
Specify automatic data set protection, e.g.

```
ALTUSER userid ADSP
```

Alter the default profile, e.g.

```
ALTDSD 'userid.RACF.MODEL.PROFILE' NOSET UACC(NONE)
```

Enter SHARE commands to define the levels of access to be authorized.

(14) After testing RACF successfully with the selected users, protect all system data sets, again using ADDSD and DELDSD commands. Issue the appropriate ALTDSD commands to define the access available to the default profiles of the groups or users associated with the system data sets and use SHARE commands to specifically protect any individual data sets that require a different level of access.
For example, most SYS1 data sets can be read by users. The default profile for group SYS at DRCS therefore specifies UACC(READ). However certain data sets required a higher level of access, such as SYS1.BRODCAST, and must have their own profiles.

(15) Educate users and induce them to define access authorities to their data sets, (e.g. by providing access reports to owners and users of data sets).

(16) Educate the group administrators and have them check and correct the users connected to the groups and their group authorities.

(17) When an appropriate period has elapsed, turn on the DSCB protect flags for all disk data sets. This can be done by generating commands as in (13) by processing a VTOC listing or catalog listing. Alter user profiles for automatic data set protection (the SEARCH command can be used to generate a CLIST). Alter the default profiles to specify UACC(NONE), again using the SEARCH command.
Delete any disk data set profiles for which no data set exists (caused during the period when specifically defined data sets did not have the DSCB bits on and therefore the profiles were not deleted when the data sets were).

APPENDIX VII

MODIFICATION TO THE RACDEF SVC

This modification permits RACDEF SVCs to be issued for data sets on volume ARCHIV, even though it is not online. This volume serial number is used by the DRCS data migration scheme to denote data sets in the archives.

The modification is to CSECT ICHRDF00, which is at MVS Rel 3.8A base level and is expressed in SMP4 format.

```
++USERMOD(LOCZ017) .
++VER(Z038) FMID(HRF1302) .
++ZAP(ICHRDF00) .
 NAME ICHRDF00
 VER 1214 4770A1FF                             BNE @RF00745
 VER 3552 00000000,00000000,00000000           *PATCH AREA*
 VER 355E 00000000,00000000,00000000           *PATCH AREA*
 REP 1214 47F0C531                             B   3552
 REP 3552 4780A1F5                             BE  1218
 REP 3556 D50581CCC543                         CLC RACFVOL,ARCHIV
 REP 355C 4780A1F5                             BE  1218
 REP 3560 4770A1FF                             BNE @RF00745
 REP 3564 C1D9C3C8C9E5                  ARCHIV DC  C'ARCHIV'
 IDRDATA LOCZ017
```

APPENDIX VIII

MODIFICATION TO OPEN FOR CREATION OF TAPE DATA SETS

This modification passes the JFCB and therefore the data set name to the RACFDEF SVC whenever a new tape data set is defined or a new volume added to an existing one.

The modification is to CSECT IFG1094F, which is at PTF UZ22357 level, and is expressed in SMP4 format.

```
++USERMOD(LOCZ014) .
++VER FMID(EDM1102) PRE(UZ22357) .
++ZAP(IFG0194F) .
 NAME IFG0194A IFG0194F
 VER 1012 4100A01C        LA 0,UCBVOLI       ADDRESS OF VOLUME
 VER 11B0 C9C6C7F0F1F9F4C6                   IFG0194F
 VER 11BA 61                                 /
 VER 11BD 61                                 /
 VER 11C0 E5E2F260D9F211C8                   VS2-R2.H
 REP 1012 47F0C1F0        B +11B8
 REP 11B8 41004064        LA 0,DXJBF         ADDRESS OF JFCB
 REP 11BC BE071001        STCM 0,7,1(1)      STORE IN INSTLN FIELD
 REP 11C0 4100A01C        LA 0,UCBVOLI       ADDRESS OF VOLUME
 REP 11C4 47F0C04E        B +1016
 IDRDATA LOCZ014
```

APPENDIX IX

MODIFICATION TO JES2

The purpose of this modification is to place the name of the JES reader that processed a job in columns 73 to 80 of the JOB card. The information is then available to SMF exit IEFUJV for validity checking. It can be used, for instance, to prevent certain users from accessing TSO, or to place constraints on which users may submit batch jobs from particular RJEs.

The modification is to module HASPRDR, which is at PTF UZ24623 level, and is expressed in SMP4 format.

```
++USERMOD(LOCSU03) .
++VER(Z038) FMID(EJE1102) PRE(UZ24623) .
++SRCUPD (HASPRDR) DISTLIB(HASPSRC) .
./ CHANGE NAME=HASPRDR,SEQFLD=747
*****    DROP  R1                  DROP DCT ADDRESSABILITY      LOCSU03 92734000
         L     R1,PCEDCT           R1 = ADDRESS OF INPUT DCT    LOCSU03 92738001
         MVC   72(8,RPI),DCTDEVN   PLACE READER NAME IN 73-80   LOCSU03 92738002
         DROP  R1                  DROP DCT ADDRESSABILITY      LOCSU03 92738005
./ ENDUP
```

APPENDIX X

MODIFICATIONS TO DADSM DURING RELEASE OF DISK SPACE

X.1 Aim

The aim of the modification is to ensure that all disk space is erased as it is freed, thereby overcoming the security problems created by residual data. The erasure is automatic and is performed as a result of a scratch request (SVC 29) and a partial release request.

X.2 Method

Both functions of DADSM have been modified to pass control to a module located in the link pack area (SCRIBBLE) to perform the actual erasure. In addition, DASDM has been altered to ensure that the disk volume is not reserved (enqueued) while the erasure is in progress, which could be for a considerable time, depending on the size of the data set.

The relevant steps currently performed by partial release are:-

(1) reserve the disk

(2) read the format 4 DSCB

(3) set the DIRF bit and rewrite the format 4 DSCB

(4) enqueue on the data set and process its format 1 DSCB, building a table of extents to be freed

(5) read and process the format 3 DSCB, if necessary, adding to the extent table

(6) delete the format 3 DSCB, if necessary, or

(7) rewrite the format 3 DSCB, if necessary

(8) rewrite the format 1 DSCB

(9) update the format 5 DSCB free space chain if no previous VTOC error

(10) reset the DIRF bit and rewrite format 4 DSCB

(11) release the disk

This logic has been changed to the following:-

(1) read the format 4 DSCB

(2) enqueue on the data set and process its format 1 DSCB, building the extent table

(3) read and process the format 3 DSCB, if necessary, adding to the extent table

(4) invoke SCRIBBLE to erase the space

(5) reserve the disk

(6) reread the format 4 DSCB

(7) set the DIRF bit and rewrite the format 4 DSCB

(8) delete the format 3 DSCB, if necessary, or

(9) rewrite the format 3 DSCB, if necessary

(10) rewrite the format 1 DSCB

(11) update the format 5 DSCB chain if no previous VTOC error

(12) reset the DIRF bit and rewrite the format 4 DSCB

(13) release the disk

This sequence ensures that the disk is not reserved during the possibly lengthy erasure while maintaining full integrity for the VTOC. In addition the erasure is performed even if the DIRF bit was originally set in the format 4 DSCB, indicating a previous VTOC error. This ensures that all unallocated areas on the disk will be clear when the VTOC is rebuilt.

A similar reorganization was made to the scratch logic. It currently is:-

(1) enqueue on the data set

(2) reserve the VTOC

(3) read the format 1 DSCB and format 4 DSCB

(4) set the DIRF bit and rewrite the format 4 DSCB

(5) process the format 1 DSCB, building a table of extents to be freed

(6) delete the format 1 DSCB by overwriting with a format 0 DSCB, reread it and then read the next DSCB in the chain (format 2 or 3 DSCB, or format 5 DSCB at the end of the chain)

(7) repeat steps (5) and (6), processing the current DSCB, overwriting it and reading the next, until the end of the chain, when the first format 5 DSCB is read instead

(8) update the format 5 DSCB free space chain if no previous VTOC error

(9) reset the DIRF bit and rewrite format 4 DSCB

(10) release the disk

This logic has been changed to the following:-

(1) enqueue on data set

(2) read the format 1 DSCB and format 4 DSCB

(3) process the format 1 DSCB, building the extent table

(4) save the address of the format 1 DSCB, read the next DSCB in the chain, if any

(5) repeat steps (3) and (4), processing the current DSCB, saving its address and reading the next, until the end of the chain

(6) invoke SCRIBBLE to erase the space

(7) reserve the disk

(8) reread the format 4 DSCB

(9) set the DIRF bit and rewrite the format 4 DSCB

(10) delete the DSCBs whose addresses have been saved, if any (by overwriting with a format 0 DSCB and read checking)

(11) delete the last DSCB in the chain and read the first format 5 DSCB

(12) update the format 5 DSCB free space chain if no previous VTOC error

(13) reset the DIRF bit and rewrite the format 4 DSCB

(14) release the disk

X.3 The SCRIBBLE program

The input to the program is documented in the listing below. The program builds its own DEB, DCB etc and uses the erase channel command to erase the data. On conclusion it writes a user GTF record (ID=100) describing the request it has just processed. For efficiency SCRIBBLE tries to avoid erasing space that is already clear. For data set types except ISAM (where all the space is erased) only the space indicated by the last TTR field of the format 1 DSCB, plus one extra track, is erased initially. The next track is then read to see if it is clear. If so, the erasure is terminated. Otherwise a further 30 tracks are erased, another read performed, and so on. (There is nothing magic about the figure of 30 tracks, and no tests have been made to determine an optimum value.) During this process the DEB protects space belonging to other users.

In addition SCRIBBLE addresses the problem of catalog contention during erasure. An Access Method Services deletion invokes SVC 29 with the catalog containing the data set held exclusively. To avoid prolonged lockouts to the catalog in such a case SCRIBBLE frees it if more than 5 tracks are being erased and re-enqueues prior to returning to SVC 29. Standard catalog management routines IGGPRPLF and IGGPRPLM are used for this. However they must be link-edited as aliases of module IGG0CLA1.

X.4 Operating characteristics

Tests indicate that about 30 tracks per second can be erased on a 3350 disk in a 'stand-alone' environment. The channel utilization in achieving this is quite small (about 3-4%), as is the CPU utilization (about 1.5 secs per 100 cylinders of 3350 space on a 3033). In practice we find that the average elapsed time per cylinder erased on a heavily loaded system (40+ TSO users, IMS, 5 or 6 batch jobs) is about 1.4 seconds. However the average time for a deletion initiated from TSO is only 0.4 seconds, and this increase in response time is not perceptible.

Only about 30% of space deleted in this installation is actually erased. The remainder is already clear. (We delete about 19000 tracks per hour, erasing about 5700 of them). The erase load is distributed fairly evenly over 17 disk drives and 4 channels. The overload is only 0.05% of the total capacity of each channel (assuming it can achieve 100%), and 0.33% of the capacity of each disk drive (again assuming a possible 100%).

X.5 Modifications to partial release

The modifications are expressed in SMP4 format. They apply to MVS Release 3.8 at PTF level 7908. PTF UZ23177 has been applied to CSECT IGG020P1. CSECTs IGG202P2 and IGG020P3 are at 3.8A base level.

```
++USERMOD(LOCZ021) .
++VER(Z038) FMID(EDM1102) PRE(UZ23177) .
++ZAP(IGG020P1) .
**** ZAP TO PARTIAL RELEASE TO ERASE FREED SPACE.
**** NOTE THAT CSECT IGG020P2 MUST BE EXPANDED BY 288 BYTES.
 NAME IGG020P1 IGG020P1                        ***** PARTIAL RELEASE ****
**** DUMMY OUT THE RESERVE ON THE VTOC
 VER 01B2 0A38                                 SVC  56 (RESERVE)
 REP 01B2 1BFF                                 SR   15,15
 VER 01B4 96C0B255                             OI   DSMADTB2,VTOCR+SMCE
 REP 01B4 18FF18FF                             LR   15,15 LR   15,15
****
****
****
**** DON'T RESET DIRF BIT OR REWRITE FMT4
 VER 01D4 9704B06E                             XI   DS4VTOCI,DIRFBIT
 REP 01D4 47F0C1EA                             B    SKIPWRT
****
****
****
++ZAP(IGG020P3) .
 NAME IGG020P1 IGG020P3
**** DON'T REWRITE FMT4 IF NOT ENQ'ED ON VTOC
 VER 006C 4110D118                             LA   1,DXIOB
 REP 006C 47F0C282                             B    PATCH AREA (+284)
 VER 0284 00000000,00000000,00000000,00000000 ***** PATCH AREA *****
 REP 0284 91C0B255                             TM   DSMADTB2,VTOCR+SMCE
 REP 0288 4780C086                             BZ   NOWRT
 REP 028C 4110D118                             LA   1,DXIOB
 REP 0290 47F0C06E                             B    +70
****
****
****
**** DON'T DEQ VTOC IF NOT ENQ'ED ON IT
 VER 010C 4110D1C0                             LA   1,ENQAREA
 REP 010C 47F0C292                             B    PATCH AREA (+294)
 VER 0294 00000000,00000000,00000000,00000000 ***** PATCH AREA *****
 REP 0294 91C0B255                             TM   DSMADTB2,VTOCR+SMCE
 REP 0298 4780C12C                             BZ   MSGTEST
 REP 029C 4110D1C0                             LA   1,ENQAREA
 REP 02A0 47F0C10E                             B    +110
****
****
****
++ZAP(IGG020P2) .
 EXPAND IGG020P2(288)
 NAME IGG020P1 IGG020P2
 VER 0350 00000000,00000000,00000000,00000000 ** PATCH AREA **
 VER 0360 00000000,00000000,00000000,00000000 ** PATCH AREA **
 VER 0370 00000000,00000000,00000000,00000000 ** PATCH AREA **
 VER 0380 00000000,00000000,00000000,00000000 ** PATCH AREA **
 VER 0390 00000000,00000000,00000000,00000000 ** PATCH AREA **
 VER 03A0 00000000,00000000,00000000,00000000 ** PATCH AREA **
 VER 03B0 00000000,00000000,00000000,00000000 ** PATCH AREA **
 VER 03C0 00000000,00000000,00000000,00000000 ** PATCH AREA **
 VER 03D0 00000000,00000000,00000000,00000000 ** PATCH AREA **
 VER 03E0 00000000,00000000,00000000,00000000 ** PATCH AREA **
 VER 03F0 00000000,00000000,00000000,00000000 ** PATCH AREA **
 VER 0400 00000000,00000000,00000000,00000000 ** PATCH AREA **
 VER 0410 00000000,00000000,00000000,00000000 ** PATCH AREA **
 VER 0420 00000000,00000000,00000000,00000000 ** PATCH AREA **
 VER 0430 00000000,00000000,00000000,00000000 ** PATCH AREA **
 VER 0440 00000000,00000000,00000000,00000000 ** PATCH AREA **
 VER 0450 00000000,00000000,00000000,00000000 ** PATCH AREA **
 VER 0460 00000000,00000000,00000000,00000000 ** PATCH AREA **
**** SAVE CURRENT DXCCW4-6 IN UNUSED PART OF FMT4. THESE CCW'S READ THE
**** FMT4
 VER 000A 91FFB24E                                TM   OUTCCHHR+K4,F3IND
 REP 000A 47F0C34E                                B    PATCH AREA (+350)
 REP 0350 D217B078D188                            MVC VTOCDSCB+24(24),DXCCW4
**** SET EXTENT NUMBER IN DADSM EXTENT TABLE FOR PROCESSING BY EXIT
 REP 0356 4250B1D9                                STC  5,EXTNUM
 REP 035A 91FFB24E                                TM   OUTCCHHR+K4,F3IND
 REP 035E 47F0C00C                                B    +0E
****
****
****
**** LINK TO SCRIBBLE EXIT BEFORE UPDATING FMT3
 VER 01DA 4100D170                                LA   0,DXCCW1
 REP 01DA 47F0C360                                B    PATCH AREA (+362)
 REP 0362 4250B1D9                                STC  5,EXTNUM
 REP 0366 4590C37C                                BAL  9,CALLEXIT
 REP 036A 4100D170                                LA   0,DXCCW1
 REP 036E 47F0C1DC                                B    +1DE
****
****
****
**** LINK TO SCRIBBLE EXIT BEFORE UPDATING FMT1
 VER 02A6 4130C301                                LA   3,NEXTXCTL
 REP 02A6 47F0C370                                B    PATCH AREA (+372)
 REP 0372 4590C37C                                BAL  9,CALLEXIT
 REP 0376 4130C301                                LA   3,NEXTXCTL
 REP 037A 47F0C2A8                                B    +2AA
****
****
****
**** THIS EXIT INVOKES SCRIBBLE AND PROCESSES THE VTOC.
**** LEAVE IF VTOC ALREADY RESERVED (IE. IF WE HAVE ALREADY BEEN THROUGH
**** HERE). THIS WILL HAPPEN IF THE DATA SET HAD BOTH A FMT1 AND FMT3,
**** WHEN THE EXIT WILL BE CALLED TWICE
 REP 037E 91C0B255                       CALLEXIT TM   DSMADTB2,VTOCR+SMCE
 REP 0382 0779                                    BNZR 9
**** DON'T INVOKE SCRIBBLE IF NO EXTENTS
 REP 0384 9500B1D9                                CLI  EXTNUM,0
 REP 0388 4780C3D2                                BE   PASSEXIT
**** ESTABLISH RETURN ADDRESS
 REP 038C 41E0C3D2                                LA   14,PASSEXIT
**** SETUP PARAMETERS FOR SCRIBBLE
 REP 0390 4170B1D8        EXTENT TABLE            LA   7,DADSMTBL
 REP 0394 5880D230        UCB ADDRESS             L    8,DXUCBADR
 REP 0398 186B            SAVE AREA               LR   6,11
 REP 039A BF88B075        TRKS/CYL                ICM  8,8,DS4DEVSZ+3
 REP 039E 41A0D064        DSNAME                  LA   10,DXJBF
 REP 03A2 BFA8C3C8        'R'                     ICM  10,8,SCRIBBLE+2
**** SIMULATE ICRES MACRO USED BY DADSM FOR TRANSFERRING CONTROL
 REP 03A6 18FB                                    LR   15,WRKAREA
 REP 03A8 900EF000                                STM  0,14,0(15)
 REP 03AC 41100020                                LA   1,X'20'
 REP 03B0 1BF1                                    SR   15,1
 REP 03B2 D20BB054C3C6                            MVC  WTGMODNM(12),SCRIBBLE
 REP 03B8 4160B054                                LA   6,WTGMODNM
 REP 03BC 58500010                                L    5,CVTPTR
 REP 03C0 58505110                            L    5,X'110'(5)
 REP 03C4 47F05014        END OF ICRES        B    20(5)
 REP 03C8 E2C3D9C9,C2C2D3C5,00000000 SCRIBBLE DC   C'SCRIBBLE',F'0'
 REP 03D4 D207B054C336               PASSEXIT MVC  WTGMODNM(8),IGG020P2
**** SAVE THE CURRENT DISK ADDRESS AND SET IT TO THE VTOC ADDRESS
 REP 03DA D207B030D138                        MVC  48(8,11),DXDAADDR
 REP 03E0 D204D13BB23B                        MVC  DXDAADDR+3(5),VTOCADR
**** SAVE CURRENT DXCCW4-6 AND SET THEM TO REREAD FMT4
 REP 03E6 D217B018D188                        MVC  24(24,11),DXCCW4
 REP 03EC D217D188B078                        MVC DXCCW4(24),VTOCDSCB+24
 REP 03F2 4110D188                            LA   1,DXCCW4
 REP 03F6 5010D128                            ST   1,IOBSIOCC
 REP 03FA 9200D19C                            MVI  DXCCW6+4,0
**** NOW RESERVE THE VTOC OF THE DISK (THIS CODE IS THE EXPANSION OF THE
**** RESERVE MACRO)
 REP 03FE D70FD1C0D1C0                        XC   ENQAREA(16),ENQAREA
 REP 0404 4110D1C0                            LA   1,ENQAREA
 REP 0408 92061001                            MVI  1(1),6
 REP 040C 96181002                            OI   2(1),24
 REP 0410 41E0C45A                            LA   14,VTOCNAME
 REP 0414 50E01004                            ST   14,4(1)
 REP 0418 58E0D230                            L    14,DXUCBADR
 REP 041C 41E0E01C                            LA   14,28(14)
 REP 0420 50E01008                            ST   14,8(1)
 REP 0424 41E0D15C                            LA   14,DXDEB+32
 REP 0428 50E0100C                            ST   14,12(1)
 REP 042C 92FFD1C0                            MVI  ENQAREA,255
 REP 0430 0A38                                SVC  56 (RESERVE)
**** INDICATE VTOC RESERVED, READ FMT4, RESET DIRF BIT AND REWRITE FMT4
**** IF NO PREVIOUS VTOC ERROR
 REP 0432 96C0B255                            OI   DSMADTB2,VTOCR+SMCE
 REP 0436 45E0C2D0                            BAL  RLINK,EXECIO
 REP 043A 9704B06E                            XI   DS4VTOCI,DIRFBIT
 REP 043E 9104B06E                            TM   DS4VTOCI,DIRFBIT
 REP 0442 4780C452                            BZ   EXITEXIT
 REP 0446 9205D198                            MVI  DXCCW6,X'05'
 REP 044A 45E0C2D0                            BAL  RLINK,EXECIO
**** RESTORE DXCCW4-6 AND CURRENT DISK ADDRESS
 REP 044E D217D188B018               EXITEXIT MVC  DXCCW4(24),24(11)
 REP 0454 D207D138B030                        MVC  DXDAADDR,48(11)
 REP 045A 07F9                                BR   9
 REP 045C E2E8E2E5,E3D6C340          VTOCNAME DC   C'SYSVTOC '
```

X.6 Modifications to scratch

The modifications are expressed in SMP4 format. They apply to MVS Release 3.8 at PTF level 7908. CSECTs IGG0290E and IGG0299A are both at 3.8A base level.

```
++USERMOD(LOCZ020) .
++VER(Z038) FMID(EDM1102) .
++ZAP(IGG0290E) .
**** ZAP TO SCRATCH TO ERASE FREED SPACE.
**** NOTE THAT CSECT IGG0299A MUST BE EXPANDED BY 336 BYTES.
 NAME IGC0002I IGG0290E                       ***** SCRATCH *****
**** DUMMY OUT THE RESERVE ON THE VTOC
 VER 0306 0A38                                SVC  56 (RESERVE)
 REP 0306 1BFF                                SR   15,15
 VER 0308 9640D300                            OI   STYPEFLG,VTOCENQ
 REP 0308 18FF18FF                            LR   15,15 LR   15,15
 VER 030C 96C0D36D                            OI   DSMADTB2,VTOCR+SMCE
 REP 030C 18FF18FF                           LR   15,15 LR   15,15
****
****
****
++ZAP(IGG0299A) .
 EXPAND IGG0299A(366)
 NAME IGC0002I IGG0299A
**** DO NOT SET THE DIRF BIT OR REWRITE THE FMT4
 VER 0166 9704D06E                           XI   DS4VTOCI,DIRFBIT
 VER 016A 9104D06E                           TM   DS4VTOCI,DIRFBIT
 VER 016E 4780C17C                           BZ   SKPWR
 VER 0172 9205D248                           MVI  CCW3,X'05'
 VER 0176 9200D24C                           MVI  CCW3+4,X'00'
 VER 017A 45E0C360                           BAL  RETURN,EXCPIO
 REP 017E 9704D06E                  SKPWR    XI   DS4VTOCI,DIRFBIT
**** BYPASS WRITING DSCB 0 OVER THE LAST DSCB AND REREADING IT. INSTEAD
**** SETUP THE CHANNEL PROGRAM TO JUST READ THE NEXT DSCB
 REP 0166 4110D278                           LA   1,CCW9
 REP 016A 5010D220                           ST   1,IOB+16
**** SAVE CCW1-CCW3 IN UNUSED PART OF FMT4. THESE CCW'S READ THE FMT4
 REP 016E D217D078D238                       MVC  VTOCDSCB+24(24),CCW1
**** SAVE THE LAST TTR AND DSORG FIELDS OF THE FMT1
 REP 0174 D202D001D122                       MVC  1(3,13),DS1LSTAR
 REP 017A D200D000D112                       MVC  0(1,13),DS1DSORG
 REP 0180 18FF                               LR   15,15
****
****
****
**** GO SAVE THE LAST DSCB ADDRESS
 VER 01E0 4780C2B2                           BZ   LASTDSCB
 REP 01E0 47F0C56A                           B    PATCH AREA (+56C)
****
****
****
**** AT END OF DSCB CHAIN BRANCH TO INVOKE SCRIBBLE
 VER 02B4 9180D06E                  LASTDSCB TM   DS4VTOCI,DOSBIT
 REP 02B4 47F0C43E                  LASTDSCB B    PATCH AREA (+440)
****
****
****
 VER 0440 00000000,00000000,00000000,00000000 ** PATCH AREA **
 VER 0450 00000000,00000000,00000000,00000000 ** PATCH AREA **
 VER 0460 00000000,00000000,00000000,00000000 ** PATCH AREA **
 VER 0470 00000000,00000000,00000000,00000000 ** PATCH AREA **
 VER 0480 00000000,00000000,00000000,00000000 ** PATCH AREA **
 VER 0490 00000000,00000000,00000000,00000000 ** PATCH AREA **
 VER 04A0 00000000,00000000,00000000,00000000 ** PATCH AREA **
 VER 04B0 00000000,00000000,00000000,00000000 ** PATCH AREA **
 VER 04C0 00000000,00000000,00000000,00000000 ** PATCH AREA **
 VER 04D0 00000000,00000000,00000000,00000000 ** PATCH AREA **
 VER 04E0 00000000,00000000,00000000,00000000 ** PATCH AREA **
 VER 04F0 00000000,00000000,00000000,00000000 ** PATCH AREA **
 VER 0500 00000000,00000000,00000000,00000000 ** PATCH AREA **
 VER 0510 00000000,00000000,00000000,00000000 ** PATCH AREA **
 VER 0520 00000000,00000000,00000000,00000000 ** PATCH AREA **
 VER 0530 00000000,00000000,00000000,00000000 ** PATCH AREA **
 VER 0540 00000000,00000000,00000000,00000000 ** PATCH AREA **
 VER 0550 00000000,00000000,00000000,00000000 ** PATCH AREA **
 VER 0560 00000000,00000000,00000000,00000000 ** PATCH AREA **
 VER 0570 00000000,00000000,00000000,00000000 ** PATCH AREA **
 VER 0580 00000000,00000000,00000000,00000000 ** PATCH AREA **
```

```
**** DON'T INVOKE SCRIBBLE IF NO EXTENTS
 REP 0440 9500D301                            CLI  EXTNUM,0
 REP 0444 4780C492                            BE   PASSEXIT
**** ESTABLISH RETURN ADDRESS
 REP 0448 41E0C492                            LA   14,PASSEXIT
**** SETUP PARAMETERS FOR SCRIBBLE
 REP 044C 4170D300       EXTENT TABLE         LA   7,DADSMTBL
 REP 0450 5880D1F8       UCB ADDRESS          L    8,WKADEB+UCBADDR
 REP 0454 186D           SAVE AREA            LR   6,13
 REP 0456 BF88D075       TRKS/CYL             ICM  8,8,DS4DEVSZ+3
 REP 045A 41A0D2D2       DSNAME               LA   10,PDSNAME
 REP 045E BFA8C486       'S'                  ICM  10,8,SCRIBBLE
 REP 0462 58B0D000       TTR,DSORG            L    11,0(13)
**** SIMULATE THE ICRES MACRO USED BY DADSM FOR TRANSFERRING CONTROL
 REP 0466 18FD                                LR   15,WRKAREA
 REP 0468 900EF000                            STM  0,14,0(15)
 REP 046C 41100020                            LA   1,X'20'
 REP 0470 1BF1                                SR   15,1
 REP 0472 D20BD054C486                        MVC  WTGMODNM(12),SCRIBBLE
 REP 0478 4160D054                            LA   6,WTGMODNM
 REP 047C 58500010                            L    5,CVTPTR
 REP 0480 58505110                            L    5,X'110'(5)
 REP 0484 47F05014       END OF ICRES         B    20(5)
 REP 0488 E2C3D9C9,C2C2D3C5,00000000 SCRIBBLE DC   C'SCRIBBLE',F'0'
 REP 0494 D207D054C426               PASSEXIT MVC  WTGMODNM(8),IGG0299A
**** SAVE THE LIST OF DSCB ADDRESSES TO BE DELETED AND CURRENT CCW1-CCW3
 REP 049A D20FD018D090                        MVC  24(16,13),VTOCDSCB+48
 REP 04A0 D217D000D238                        MVC  0(24,13),CCW1
**** SET CCW1-CCW3 TO REREAD FMT4
 REP 04A6 D217D238D078                        MVC  CCW1(24),VTOCDSCB+24
 REP 04AC 9200D24C                            MVI  CCW3+4,X'00'
 REP 04B0 D204D34ED344                        MVC  INCCHHR,VTOCADR
 REP 04B6 D204D233D34E                        MVC  SEEK+3(5),INCCHHR
 REP 04BC 41E0D238                            LA   14,CCW1
 REP 04C0 50E0D220                            ST   14,IOB+16
**** NOW RESERVE THE VTOC OF THE DISK (THIS CODE IS THE EXPANSION OF THE
**** RESERVE MACRO)
 REP 04C4 D70FD150D150                        XC   ENQAREA(16),ENQAREA
 REP 04CA 4110D150                            LA   1,ENQAREA
 REP 04CE 92061001                            MVI  1(1),6
 REP 04D2 96181002                            OI   2(1),24
 REP 04D6 41E0C562                            LA   14,VTOCNAME
 REP 04DA 50E01004                            ST   14,4(1)
 REP 04DE 58E0D1F8                            L    14,WKADEB+UCBADDR
 REP 04E2 41E0E01C                            LA   14,28(14)
 REP 04E6 50E01008                            ST   14,8(1)
 REP 04EA 41E0D1F8                            LA   14,WKADEB+UCBADDR
 REP 04EE 50E0100C                            ST   14,12(1)
 REP 04F2 92FFD150                            MVI  ENQAREA,255
 REP 04F6 0A38                                SVC  56 (RESERVE)
**** INDICATE VTOC RESERVED, READ FMT4, RESET DIRF BIT AND REWRITE FTM4
**** IF NO PREVIOUS VTOC ERROR
 REP 04F8 9640D300                            OI   STYPEFLG,VTOCENQ
 REP 04FC 96C0D36D                            OI   DSMADTB2,VTOCR+SMCE
 REP 0500 45E0C360                            BAL  RETURN,EXCPIO
 REP 0504 9704D06E                            XI   DS4VTOCI,DIRFBIT
 REP 0508 9104D06E                            TM   DS4VTOCI,DIRFBIT
 REP 050C 4780C516                            BZ   SKIPWRT
 REP 0510 9205D248                            MVI  CCW3,X'05'
 REP 0514 45E0C360                            BAL  RETURN,EXCPIO
 REP 0518 9704D06E                    SKIPWRT XI   DS4VTOCI,DIRFBIT
```

```
**** RESTORE CCW1-CCW3 WITH COMMANDS TO WRITE DSCB 0
 REP 051C D217D238D000                          MVC  CCW1(24),0(13)
**** GET NUMBER OF DSCB'S THAT SHOULD HAVE ALREADY BEEN DELETED. RETURN
**** TO MAINLINE IF NONE
 REP 0522 4820D302                              LH   2,DADSMTBL+2
 REP 0526 1222                                  LTR  2,2
 REP 0528 4780C55A                              BZ   NONEDEL
**** SAVE CURRENT OUTCCHHR
 REP 052C D204D028D353                          MVC  40(5,13),OUTCCHHR
**** LIST OF DSCB ADDRESSES TO DELETE
 REP 0532 4130D018                              LA   3,24(13)
 REP 0536 94BFD264                              NI   CCW6+4,X'BF'
**** WRITE A DSCB 0 OVER EACH OF THE DSCB'S AND READ CHECK
 REP 053A D204D3533000                LOOP      MVC  OUTCCHHR,0(3)
 REP 0540 41303008                              LA   3,8(3)
 REP 0544 D204D233D353                          MVC  SEEK+3(5),OUTCCHHR
 REP 054A 45E0C360                              BAL  RETURN,EXCPIO
 REP 054E 4620C538                              BCT  2,LOOP
**** INDICATE COMMAND CHAINING. THERE IS STILL 1 DSCB TO BE DELETED,
**** READ CHECKED AND THEN A DSCB 5 OR 6 TO BE READ USING THE UNMODIFIED
**** CHANNEL PROGRAM
 REP 0552 9640D264                              OI   CCW6+4,X'40'
**** RESTORE THE CURRENT OUTCCHHR
 REP 0556 D204D353D028                          MVC  OUTCCHHR(5),40(13)
 REP 055C 9180D06E                    NONEDEL   TM   DS4VTOCI,DOSBIT
 REP 0560 47F0C2B6                              B    +2B8
 REP 0564 E2E8E2E5,E3D6C340           VTOCNAME DC   C'SYSVTOC '
****
****
****
 REP 056C 4780C2B2                              BZ   LASTDSCB
**** SAVE THE CCHHR OF THE LAST DSCB IN AN UNUSED PART OF THE FMT4
**** FOR LATER DELETION
 REP 0570 4110D090                              LA   1,VTOCDSCB+48
 REP 0574 48F0D302                              LH   WORKREG,DADSMTBL+2
 REP 0578 89F00003                              SLL  WORKREG,3
 REP 057C 4111F000                              LA   1,0(1,WORKREG)
 REP 0580 D2041000D353                          MVC  0(5,1),OUTCCHHR
 REP 0586 D204D353D34E                          MVC  OUTCCHHR(5),INCCHHR
 REP 058C 47F0C1E2                              B    ZEROUT
```

X.7 SCRIBBLE program listing

```
SCRIBBLE START 0
*
* THIS ROUTINE IS CALLED FROM DASDM PARTIAL RELEASE (IGG020P2) AND
* DADSM SCRATCH (IGG0299A) TO ERASE SPACE BEING FREED BEFORE IT IS
* PUT BACK ON THE FMT5 FREE SPACE LIST.
* ON ENTRY THE FOLLOWING INFORMATION IS AVAILABLE -
*     REG 6 HAS THE ADDRESS OF A SAVE AREA
*     REG 7 HAS THE ADDRESS OF THE DADSM EXTENT TABLE
*     REG 8 HAS THE NUMBER OF TRACKS PER CYLINDER FOR THE DEVICE IN
*           BYTE 0 AND THE UCB ADDRESS IN BYTES 1 TO 3
*     REG 10 HAS 'S' IN BYTE 0 IF CALLED FROM SCRATCH OR 'R' IF CALLED
*           FROM PARTIAL RELEASE AND HAS THE DATASET NAME ADDRESS IN
*           BYTES 1 TO 3
*     REG 11 HAS THE DATASET ORGANIZATION FROM THE DS1DSORG FIELD IN
*           BYTE 0 AND THE TTR OF THE LAST BLOCK FROM THE DS1LSTAR
*           FIELD IN BYTES 1 TO 3 (FOR A SCRATCH REQUEST ONLY)
*
```

```
          USING *,12
          STM   0,14,0(6)           SAVE THE REGISTERS
          LR    13,6                ADDRESS OF CALLER'S SAVE AREA
          LR    12,15
          SR    15,15
*
* TEST FOR NON-ZERO PARAMETERS
*
          LTR   7,7                 EXTENT TABLE
          BZ    BADPARM             ERROR
          CLM   8,8,=F'0'           TRACKS PER CYLINDER
          BE    BADPARM             ERROR
          CLM   8,7,=F'0'           UCB ADDRESS
          BE    BADPARM             ERROR
          CLM   10,7,=F'0'          DATASET NAME ADDRESS
          BE    BADPARM             ERROR
*
* CALCULATE LENGTH OF WORK AREA AND GET IT
*
          USING DADSMTBL,7
          SR    3,3
          IC    3,EXTNUM            NUMBER OF DATA EXTENTS
          C     3,=F'16'            ENSURE NOT MORE THAN 16
          BH    BADPARM             ERROR
          LA    5,LENDEBEX          LENGTH OF EXTENT SECTION IN DEB
          LA    6,ENDGET-WORK       BASIC WORK AREA LENGTH (1 EXTENT)
          LTR   3,3                 ARE THERE ANY EXTENTS ?
          BZ    RETURN              NO - GO BACK
          BCTR  3,0                 ALREADY ACCOUNTED FOR 1 EXTENT
          MR    2,5
          AR    3,6                 WORK AREA LENGTH
          LA    4,OUTIOVEC-WORK     LENGTH OF NON-DEB WORK AREA
          LR    5,3
          SR    5,4                 LENGTH OF DEB
          SRL   5,3                 NUMBER OF DOUBLE WORDS IN DEB
          GETMAIN RC,LV=(3),SP=230,RELATED=WORK
          LTR   15,15               OK ?
          BNZ   GETERROR            NO - TERMINATE
          LR    9,1                 ADDRESS OF WORK AREA
          USING WORK,9
*
* ZERO WORK AREA
*
          LR    6,3                 LENGTH
REPZERO   LA    4,256               256 BYTES AT A TIME
          CR    4,6                 REMAINING AREA LESS THAN 256 ?
          BNH   ZERO                NO
          LR    4,6                 YES - ZERO ONLY THIS AMOUNT
ZERO      SR    6,4                 DECREASE AREA REMAINING
          BCTR  4,0                 DECREMENT FOR EX
          EX    4,ZEROUT            ZERO
          LA    1,256(1)            UPDATE WORK AREA LOCATION
          LTR   6,6                 ANY AREA STILL TO BE DONE ?
          BNZ   REPZERO             YES
          STH   3,WORKLEN           SAVE AREA LENGTH FOR FREEMAIN
          STCK  TIMEIN              REMEMBER TIME OF ENTRY
          DROP  7
          ST    7,R7SAVE            SAVE REG 7
          ST    8,R8SAVE            SAVE REG 8
          ST    10,R10SAVE          SAVE REG 10
          ST    11,R11SAVE          SAVE REG 11
```

```
         EJECT
* CONSTRUCT IOB, CCW'S, DCB AND DEB
         L     4,16                GET ADDRESS OF TCB - START WITH CVT
         L     4,0(4)
         L     4,4(4)
         ST    4,TCBADDR           SAVE IN WORK AREA
         LA    3,MYECB             BUILD IOB
         ST    3,ECBA              ECB ADDRESS
         LA    3,CCW
         ST    3,CCWA              COMMAND ADDRESS
         MVI   FL1,X'C2'           SET DATA,COMMAND CHAINING,UNRELATED
         MVC   CCW(LENCCW),CCWD    INITIALIZE CHANNEL PROGRAM
         LA    3,MYSEEK+3          SEEK ADDRESS
         STCM  3,7,SEARCH+1        STORE IN SEARCH RO CCW
         LA    3,SEARCH            SEARCH CCW ADDRESS
         STCM  3,7,TIC+1           STORE IN TIC CCW
         LA    3,SDATA             DATA ADDRESS
         STCM  3,7,ERASECKD+1      STORE IN ERASE CCW
         LA    3,LENSDATA          DATA LENGTH
         STH   3,ERASECKD+6        STORE IN ERASE CCW
         LA    3,OUTDCB
         ST    3,DCBA              DCB ADDRESS
         MVC   OUTDCB(LENDCBDB),DCBDEB PLACE DCB AND DEB IN WORK AREA
         STC   5,DEBLEN            STORE DEB LENGTH IN PREFIX
         LA    3,OUTDEB            ADDRESS OF DEB
         ST    3,DCBDEBAD          STORE IN DCB
         LA    3,OUTDCB            ADDRESS OF DCB
         STCM  3,7,DEBDCBB         STORE IN DEB
         LA    3,OUTIOVEC          ADDRESS OF APPENDAGE LIST
         STCM  3,7,DEBAPPB         STORE IN DEB
         L     4,R8SAVE            UCB ADDRESS
         MVC   DCBDEVT,18(4)       EXTRACT DEVICE TYPE FOR DCB
         OC    DCBDEVT,19(4)
         L     3,16                CVT
         L     3,64(3)             ADDR OF I/O DEVICE CHAR TABLE
         SR    1,1                 CLEAR 1
         IC    1,19(4)             DEVICE CODE
         IC    1,0(1,3)            CONSTRUCT ADDRESS OF ENTRY IN ...
         LA    3,0(1,3)            DEVICE CHARACTERISTICS TABLE
         ST    3,DCBDVTBL          STORE IN DCB
         USING DADSMTBL,5
         L     5,R7SAVE            ADDRESS OF DADSM EXTENT TABLE
         MVC   DEBNMEXT,EXTNUM     NUMBER OF DATA EXTENTS
         MVC   DEBTCBAD,TCBADDR    MOVE TCB ADDRESS TO DEB
         EJECT
* FILL IN THE EXTENT DESCRIPTIONS IN THE DEB
         SR    3,3
         SR    14,14
         IC    3,EXTNUM            NUMBER OF EXTENTS
         SR    2,2
         IC    2,R8SAVE            NUMBER OF TRACKS PER CYLINDER
         LA    4,ENTRIES           POINT AT FIRST EXTENT IN SCRTHWKA
         LA    10,DEBDVMOD         POINT AT FIRST EXTENT ENTRY IN DEB
         USING DEBDVMOD,10
EXTFILL  EQU   *
         MVI   DEBDVMOD,X'18'      FILE MASK
         MVC   DEBUCBA(3),R8SAVE+1 UCB ADDRESS
         LH    7,0(4)              EXTENT START TRACK
         LR    11,7                SAVE
         SR    6,6
         DR    6,2                 DIVIDE BY TRACKS PER CYLINDER
         STH   7,DEBSTRCC          STORE START CYLINDER IN DEB
         STH   6,DEBSTRHH          STORE START TRACK IN DEB
         LH    7,2(4)              EXTENT END TRACK +1
         LR    8,7                 SAVE
         SR    8,11                TRACKS IN EXTENT
         BCTR  7,0                 EXTENT END TRACK
         SR    6,6
         DR    6,2                 DIVIDE BY TRACKS PER CYLINDER
         STH   7,DEBENDCC          STORE END CYLINDER IN DEB
         STH   6,DEBENDHH          STORE END TRACK IN DEB
         CLC   DEBSTRCC(4),=F'0'   PROTECT TRACK 0
         BE    BADEXT              ERROR
         CLC   DEBSTRCC(4),DEBENDCC ENSURE EXTENT IS VALID
         BH    BADEXT              ERROR
         STH   8,DEBNMTRK          STORE EXTENT SIZE IN DEB
         AR    14,8                ACCUMULATE TRACKS ALLOCATED
         LA    10,LENDEBEX(10)     POINT AT NEXT EXTENT ENTRY IN DEB
         LA    4,4(4)              POINT AT NEXT EXTENT IN SCRTHWKA
         BCT   3,EXTFILL           GO PROCESS NEXT EXTENT
         MVC   0(4,10),=X'00010001' INDICATE 1ST AND ONLY VOLUME
         LR    8,14                TRACKS ALLOCATED
         DROP  10
         DROP  5
         EJECT
* ADD THE DEB TO THE DEB QUEUE AND CHECK IT
         L     3,TCBADDR           TCB ADDRESS
         OC    DEBPROTG(1),28(3)   STORE PROTECTION KEY IN DEB
         L     4,8(3)              DEB QUEUE
         LR    6,4                 SAVE DEB ADDRESS
         BZ    NODEB               NO DEB CURRENTLY QUEUED
         O     6,DEBDEBB
         ST    6,DEBDEBB           POINT TO CURRENT DEB FROM OUR'S
NODEB    LA    5,OUTDEB            ADDRESS OF OUR DEB
         MODESET EXTKEY=ZERO,SAVEKEY=(2)
         ST    5,8(3)              STORE IN TCB
         MODESET KEYADDR=(2)
         DEBCHK OUTDCB,TYPE=ADD,AM=EXCP
         LTR   15,15               DEB CHECK OK ?
         BNZ   BADDEB              NO
         EJECT
* CHECK THE LAST TTR VALUE FOR SCRATCH REQUESTS
* REG 8 HAS THE NUMBER OF TRACKS ALLOCATED
         CLI   R10SAVE,C'S'        SCRATCH REQUEST ?
         BNE   CHECK2ND            NO
         TM    R11SAVE,X'80'       ISAM ?
         BZ    DSORGOK             NO
         LA    11,0                ERASE ALL TRACKS IF ISAM
         B     CHECKDEQ            CHECK IF CATALOG DEQ IS REQUIRED
DSORGOK  L     11,R11SAVE          GET TTR OF LAST BLOCK
         LA    11,0(11)            ZERO DS1DSORG BYTE
         SLL   8,8                 SHIFT TRACKS ALLOCATED FOR COMPARE
         CR    11,8                COMPARE TRACKS USED WITH ALLOCATED
         BL    TTROK               TTR IS VALID
         LA    11,0                ERASE WHOLE DATASET IF TTR INVALID
         B     CHECKDEQ            CHECK IF CATALOG DEQ IS REQUIRED
TTROK    SRL   8,8                 SHIFT TRACKS ALLOCATED BACK
         LTR   11,11               IS TTR ZERO ?
         BZ    CHECK1ST            YES - DATASET PROBABLY EMPTY OR VSAM
         SRL   11,8                GET TT ONLY IN REG 11
         LA    11,3(11)            SET UP TO ERASE TT+2 TRACKS (ALLOW 1
         B     COMPSIZE            EXTRA IN CASE EOF ON NEXT TRACK)
```

```
CHECK1ST LA    11,1                SET UP TO CHECK IF 1ST TRACK EMPTY
         B     COMPSIZE            GO CHECK DATASET SIZE
CHECK2ND LA    11,2                CHECK 2ND TRACK (IN CASE EOF ON 1ST)
COMPSIZE CR    8,11                COMPARE WITH TRACKS ALLOCATED
         BH    CHECKDEQ            MORE THAN THE ONE TO BE READ
         LA    11,0                DON'T BOTHER TO READ - JUST WRITE
         EJECT
* DELETIONS OCCURING AS A RESULT OF A REQUEST TO ACCESS METHOD
* SERVICES (AMS) ENTER SCRIBBLE WITH THE OS VSAM CATALOG HELD WITH AN
* EXCLUSIVE ENQ. TO AVOID PROLONGED LOCKOUTS OF THE CATALOG FOR LARGE
* DELETIONS IT IS DEQ'ED PRIOR TO THE ERASURE AND RE-ENQ'D AFTER.
* THE CATALOG MANAGEMENT ROUTINES IGGPRPLF AND IGGPRPLM ARE USED TO
* DEQ AND ENQ THE CATALOG RESPECTIVELY. THEY ALSO CAUSE EXTRA OVERHEAD
* RELATED TO FREEING AND REACQUIRING BUFFERS ETC.
* BOTH ROUTINES EXPECT THE ADDRESS OF THE CATALOG COMMUNICATIONS AREA
* TO BE IN REG 11 AND THE ADDRESS OF THE NEXT AVAILABLE 3 WORD SAVE
* AREA FROM THE CCA IN REG 13 AND THEY DESTROY ALL REGISTERS EXCEPT
* 11 TO 14.
* TO DETERMINE IF THIS IS AN AMS REQUEST WE NEED TO SEE IF SVC 29
* (DADSM SCRATCH) WAS INVOKED BY SVC 26 (CATALOG MANAGEMENT). IF SO
* THE REGS REQUIRED (11 AND 13) CAN BE OBTAINED FROM THE SAVE AREA OF
* THE APPROPRIATE SVRB. TO DO THIS THE RB CHAIN MUST BE TRACED. THE
* INTERRUPT CODE THAT CAUSED THE CREATION OF THE CURRENT RB IS STORED
* IN THE NEXT RB IN THE CHAIN, WHILE THE REGISTER CONTENTS WHEN IT
* RELINQUISHED CONTROL ARE IN THE PREVIOUS RB IN THE CHAIN.
* THE LINK SVC IS USED TO TRANSFER CONTROL TO IGGPRPLF AND IGGPRPLM
* AND THIS REQUIRES BOTH TO BE DEFINED AS ALIASES OF IGG0CLA1.
*
CHECKDEQ DS    0H                  CHECK IF CATALOG DEQ IS NECESSARY
         LR    2,8                 SAVE TRACKS ALLOCATED
         CLI   R10SAVE,C'S'        SCRATCH REQUEST ?
         BNE   ERASE               NO - DEQ NOT REQUIRED
         LTR   11,11               ENTIRE DATASET BEING ERASED ?
         BZ    CHECKSIZ            YES
         LR    8,11                INITIAL NO. OF I/O'S TO BE DONE
CHECKSIZ C     8,=F'5'             MORE THAN 5 I/O'S ?
         BNH   ERASE               NO - DON'T BOTHER WITH DEQ
         BAL   3,DEQCAT            PERFORM DEQ IF AN AMS REQUEST
         B     ERASE               START ERASURE
*
*
* THIS ROUTINE TESTS FOR AN AMS REQUEST AND FREES THE CATALOG IF SO
*
DEQCAT   DS    0H
         L     14,TCBADDR          ADDRESS OF TCB
         LR    7,14                SAVE
         L     14,0(14)            ADDRESS OF 1ST RB IN CHAIN
TEST29   LR    15,14
         S     15,=F'2'            ADDRESS OF INTERRUPT CODE
         CLC   0(2,15),=H'29'      LOOK FOR INTERRUPT CODE OF 29
         BNE   NEXTRB              NOT THIS ONE
         TM    10(7),X'C0'         WAS IT SVC 29 (CHAINED SVRB) ?
         BO    FOUND29             YES
NEXTRB   TM    11(14),X'80'        DOES THIS RB POINT BACK TO TCB ?
         BO    LASTRB              YES - NOT AN AMS REQUEST
         LR    7,14                NO - SAVE ADDRESS OF THIS RB
         L     14,28(14)           POINT TO NEXT RB
         B     TEST29              REPEAT SEARCH FOR SVC 29
FOUND29  DS    0H                  HAVE FOUND SVC 29
         TM    11(14),X'80'        DOES THIS RB POINT BACK TO TCB ?
         BO    LASTRB              YES - NOT CALLED FROM SVC 26
         L     1,28(14)          GET ADDRESS OF NEXT RB
         S     1,=F'2'           ADDRESS OF INTERRUPT CODE
         CLC   0(2,1),=H'26'     LOOK FOR INTERRUPT CODE OF 26
         BNE   LASTRB            NOT FOUND
         TM    10(14),X'C0'      WAS IT SVC 26 (CHAINED SVRB) ?
         BNO   LASTRB            NO
         L     15,76(7)          CONTENTS OF REG 11 FROM SVRB
         CLC   0(2,15),=X'ACCA'  DOES IT POINT TO THE CCA ?
         BNE   LASTRB            NO
         STM   2,13,SAVE         SAVE REGS
         LR    11,15             ADDRESS OF CCA FOR IGGPRPLF
         L     13,84(7)          ADDRESS OF CCA SAVE AREA
         ST    11,CCA            SAVE CCA ADDRESS FOR IGGPRPLM
         ST    13,CCASAVE        SAVE CCA SAVE AREA ADDRESS
* SIMULATE THE LINK MACRO TO INVOKE IGGPRPLF TO FREE CATALOG
         CNOP  0,4
         BAL   15,*+20           BRANCH AROUND CONSTANTS
         DC    A(*+8)            ADDRESS OF PARM LIST
         DC    A(0)              DCB ADDRESS PARAMETER
         DC    CL8'IGGPRPLF'     EP PARAMETER
         LR    12,9              SAVE BASE (REG 12 NOT DESTROYED)
         SVC   6                 ISSUE LINK SVC
         LR    9,12              RESTORE WORK AREA BASE
         LM    2,13,SAVE         RESTORE REGISTERS
         MVC   DEQCNT,=H'1'      INDICATE DEQ PERFORMED
LASTRB   DS    0H
         BR    3                 RETURN TO CALLER
*
*
* THIS ROUTINE INVOKES IGGPRPLM TO RESERVE THE CATALOG
*
ENQCAT   DS    0H
         STM   2,13,SAVE         SAVE REGS
         L     11,CCA            CCA ADDRESS
         L     13,CCASAVE        CCA SAVE AREA ADDRESS
* SIMULATE THE LINK MACRO TO INVOKE IGGPRPLM TO RESERVE CATALOG
         CNOP  0,4
         BAL   15,*+20           BRANCH AROUND CONSTANTS
         DC    A(*+8)            ADDRESS OF PARM LIST
         DC    A(0)              DCB ADDRESS PARAMETER
         DC    CL8'IGGPRPLM'     EP PARAMETER
         LR    12,9              SAVE BASE (REG 12 NOT DESTROYED)
         SVC   6                 ISSUE LINK SVC
         LR    9,12              RESTORE WORK AREA BASE
         LM    2,13,SAVE         RESTORE REGS
         BR    3                 RETURN TO CALLER
         EJECT
* ERASE DATA
* REG 2 CONTAINS THE NUMBER OF TRACKS ALLOCATED.
* REG 11 CONTAINS THE NUMBER OF TRACKS+1 TO BE ERASED INITIALLY. WHEN
* THIS HAS BEEN DONE THE NEXT TRACK IS READ TO SEE IF IT IS ALREADY
* ERASED. IF SO THE REMAINDER OF THE DATASET IS ASSUMED TO BE CLEAR
* AND WILL NOT BE ERASED. HOWEVER IF THE TRACK READ IS NOT EMPTY A
* FURTHER 30 TRACKS WILL BE ERASED AND THE NEXT READ ETC.
*
ERASE    MVC   SDATA(LENSDATA),SDATAD
         L     8,=X'00000000'    INITIAL TTRN
         SR    10,10             NUMBER OF TRACKS READ
EXCP     L     1,DCBDEBAD        DEB ADDR
         LR    0,8
         LR    7,9               SAVE BASE (7 NOT DSTRYD)
         STM   2,13,SAVE          SAVE REGS
         LA    2,MYSEEK
         L     15,16              CVT
         L     15,28(15)          TTR CONVERT ROUTINE
         BALR  14,15
         LR    9,7
         LM    2,13,SAVE          RESTORE REGS
         LTR   15,15
         BNZ   CLOSE              END OF ALLOCATED EXTENTS
         XR    3,3
         ST    3,MYECB            CLEAR ECB
         BCT   11,REISSUE         ERASE THE TRACK IF NOT DUE FOR READ
*
* NOW PERFORM THE READ TO SEE IF THE REST OF THE DATASET IS CLEAR
         LA    10,1(10)           INCREMENT TRACKS READ
         MVI   ERASECKD,X'1E'     READ CKD CHANNEL COMMAND
         EXCP  MYIOB              READ THE TRACK
         LA    3,MYECB
         WAIT  1,ECB=(3)          WAIT FOR READ TO COMPLETE
         CLI   MYECB,X'41'        EXPECT ERROR IF TRACK EMPTY
         BNE   ERMORE             NO ERROR - MUST CONTAIN DATA
         CLC   CSW+4(2),=X'0E00'  EXPECT UNIT CHECK ALSO
         BNE   ERMORE             NO - PROBABLY CONTAINS EOF
         CLC   SENSE,=H'8'        MUST BE NO RECORD FOUND CONDITION
         BNE   ERMORE             NO
         B     CLOSE              TRACK IS EMPTY - END ERASE
ERMORE   MVI   ERASECKD,X'11'     RESET ERASE CCW
         XC    MYECB,MYECB        CLEAR ECB
         LA    11,30              SET TO ERASE 30 MORE TRACKS
         CLI   R10SAVE,C'S'       SCRATCH REQUEST ?
         BNE   REISSUE            NO - CATALOG DEQ NOT REQUIRED
         CLC   DEQCNT,=H'0'       CATALOG ALREADY DEQUED ?
         BNE   REISSUE            YES
         SLL   2,16               SHIFT TRACKS ALLOCATED
         SR    2,8                NUMBER OF TRACKS REMAINING
         SRL   2,16               SHIFT BACK
         C     2,=F'5'            MORE THAN 5 STILL TO DO ?
         BNH   REISSUE            NO
         BAL   3,DEQCAT           YES - GO DEQ CAT BEFORE ERASING MORE
* END OF READ LOGIC
*
REISSUE  DS    0H
         MVC   CCHH,MYSEEK+3      MOVE SEEK ADDRESS TO COUNT FIELD
         EXCP  MYIOB              WRITE CRAP ON DATASET
         LA    3,MYECB
         WAIT  1,ECB=(3)
         CLI   MYECB,X'44'
         BE    REISSUE
         CLI   MYECB,X'7F'
         BNE   BADEXCP
         A     8,=X'00010000'     INCREMENT RELATIVE TRACK
         B     EXCP
CLOSE    DS    0H                 SPACE ERASED SUCCESSFULLY
         SR    2,2                ZERO RETURN CODE
         B     PURGEDEB           GO REMOVE DEB
         EJECT
BADPARM  WTO   'SCRIBBLE - ERROR IN INPUT, SPACE NOT ERASED',          X
               ROUTCDE=(9),DESC=(3)
         LA    15,13              ERROR CODE
         B     RETURN
         SPACE 4
```

```
GETERROR WTO   'SCRIBBLE - ERROR IN GETMAIN, SPACE NOT ERASED',        X
               ROUTCDE=(9),DESC=(3)
         LA    15,12               ERROR CODE
         B     RETURN
         SPACE 4
BADEXT   WTO   'SCRIBBLE - ERROR IN EXTENT LIST, SPACE NOT ERASED',     X
               ROUTCDE=(9),DESC=(3)
         LA    15,14               ERROR CODE
         B     FREE
         SPACE 4
BADDEB   WTO   'SCRIBBLE - DEB CHECK FAILED, SPACE NOT ERASED',        X
               ROUTCDE=(9),DESC=(3)
         LA    2,15                RETURN CODE
         B     UNCHAIN             REMOVE FROM TCB DEB QUEUE
         SPACE 4
BADEXCP  WTO   'SCRIBBLE - ERROR IN CHANNEL PROGRAM, SPACE MAY NOT HAVEX
                BEEN ERASED',ROUTCDE=(9),DESC=(3)
         LA    2,8                 RETURN CODE
         SPACE 4
PURGEDEB DS    0H
         DEBCHK OUTDEB,TYPE=PURGE
         LTR   15,15               ERROR ?
         BZ    UNCHAIN             NO
         WTO   'SCRIBBLE - DEB PURGE FAILED, BUT SPACE ERASED',        X
               ROUTCDE=(9),DESC=(3)
         LA    2,1                 RETURN CODE
         SPACE 4
UNCHAIN  EQU   *
         SR    4,4
         ICM   4,7,DEBDEBB+1       GET NEXT DEB ADDRESS
         L     3,TCBADDR           TCB ADDRESS
         LR    5,2                 SAVE REG 2
         MODESET EXTKEY=ZERO,SAVEKEY=(2)
         ST    4,8(3)              STORE NEXT DEB ADDRESS ON TCB QUEUE
         MODESET KEYADDR=(2)
         LR    2,5                 RESTORE REG 2
FREE     BAL   3,GTWRITE           WRITE GTF RECORD
         CLC   DEQCNT,=H'0'        WAS CATALOG DEQUED ?
         BE    WORKFREE            NO
         BAL   3,ENQCAT            YES - ENQ ON THE CATALOG AGAIN
WORKFREE LH    3,WORKLEN           GET WORK AREA LENGTH
         FREEMAIN RC,LV=(3),SP=230,A=(9),RELATED=WORK
         LTR   15,15               ERROR ?
         BZ    GETCODE             NO
         WTO   'SCRIBBLE - ERROR IN FREEMAIN, BUT SPACE ERASED',        X
               ROUTCDE=(9),DESC=(3)
         LA    2,2                 RETURN CODE
GETCODE  LR    15,2                SET RETURN CODE IN REG 15
*
* THE POSSIBLE RETURN CODES ARE
*  0 - SPACE ERASED SUCCESSFULLY
*  1 - SPACE ERASED BUT DEB PURGE FAILED
*  2 - SPACE ERASED BUT FREEMAIN FAILED
*  8 - ERROR IN CHANNEL PROGRAM AND SOME SPACE POSSIBLY NOT ERASED
* 12 - ERROR IN GETMAIN AND SPACE NOT ERASED
* 13 - ERROR IN PARAMETER INPUT AND SPACE NOT ERASED
* 14 - ERROR IN EXTENT LIST AND SPACE NOT ERASED
* 15 - DEB CHECK FAILED AND SPACE NOT ERASED
         SPACE 4
RETURN   LM    0,14,0(13)          RESTORE REGISTERS
         BR    14                  AND RETURN
         SPACE 4
APPEND   BR    14                  APPENDAGE ROUTINES
         EJECT
GTWRITE  DS    0H                  ROUTINE TO FORMAT AND WRITE GTF
         MVC   GTIMEIN,TIMEIN      PLACE TIME OF ENTRY IN GTF RECORD
         STCK  GTIMEOUT            PLACE TIME OF EXIT IN GTF RECORD
         STCM  8,12,GTNERASE       PLACE TRACKS ERASED IN GTF RECORD
         STCM  10,3,GTNREAD        PLACE TRACKS READ IN GTF RECORD
         MVC   GTNDEQ,DEQCNT       PLACE CAT DEQ/ENQ COUNT IN GTF RECORD
         SR    7,7
         IC    7,DEBNMEXT          NUMBER OF DATA EXTENTS
         STH   7,GTNMEXT           SAVE IN GTF RECORD
         MVC   GTCALLER,R10SAVE    SET CALLER CODE
         STC   2,GTCOMP            SET COMPLETION CODE
         L     10,R10SAVE          ADDRESS OF DSNAME
         MVC   GTDSN,0(10)         MOVE DSN TO GTF RECORD
         L     8,R8SAVE            ADDRESS OF UCB
         MVC   GTVOL,28(8)         MOVE VOLUME TO GTF RECORD
         LR    10,7                NUMBER OF EXTENTS
         LA    4,GTEXTS            ADDRESS OF 1ST EXTENT IN GTF RECORD
         LA    8,DEBSTRCC          ADDRESS OF 1ST EXTENT IN DEB
MOVEXT   MVC   0(10,4),0(8)        MOVE 10-BYTE EXTENT FROM DEB TO GTF
         LA    4,10(4)             NEXT GTF EXTENT DESCRIPTION
         LA    8,16(8)             NEXT DEB EXTENT DESCRIPTION
         BCT   10,MOVEXT           MOVE NEXT EXTENT
         LA    4,10                LENGTH OF EACH GTF EXTENT
         MR    6,4                 TOTAL LENGTH OF GTF EXTENTS
         LA    7,GTEXTS-GTREC(7)   TOTAL LENGTH OF GTF RECORD
         LA    8,GTREC             ADDRESS OF GTF RECORD
         MVC   GTF(LENGTMAC),GTFMAC INITIALIZE LIST FORM OF MACRO
         GTRACE MF=(E,GTF),ID=100,DATA=(8),LNG=(7),PAGEIN=YES WRITE GTF
         BR    3                   RETURN
GTFMAC   GTRACE MF=L
LENGTMAC EQU   *-GTFMAC
         EJECT
SECTOR   DC    X'00'
CCWD     CCW   X'23',SECTOR,X'60',1  SET SECTOR FOR HA
         CCW   X'31',0,X'40',5     SEARCH FOR R0
         CCW   X'08',0,0,0         TIC*-8
         CCW   X'11',0,X'60',0     ERASE
         CCW   X'03',0,X'20',5     NO-OP
LENCCW   EQU   *-CCWD
SDATAD   DS    0H
         DS    XL4   SAME AS IOBCCHH
         DC    X'0100'   R=1, KL=0
LEN      DC    AL2(L'DATA)
DATA     DC    C'SCRIBBLE'
LENSDATA EQU   *-SDATAD
ZEROUT   XC    0(0,1),0(1)
         EJECT
DCBDEB   DS    0F                  DCB FOR DATA BEING ERASED
         DS    17X'00'
         DC    X'00'
         DC    2X'00'
         DC    F'1'
         DC    H'0'
         DC    X'4000'             PS
         DC    F'1'
         DC    X'06000001'
         DC    X'C0000000'
         DC    H'0'
         DC    BL2'1101000000001000'
         DC    A(0)
         DC    X'9200'
         DC    BL2'1101000000001000'
         DC    5F'0'
         DS    0H                  DEB PREFIX
         DC    A(APPEND)
         DC    A(APPEND)
         DC    A(APPEND)
         DC    A(APPEND)
         DC    A(APPEND)
         DC    3F'0'
         DC    X'00000000'         LENGTH OF DEB IN DOUBLE WORDS
         DS    0F
         DC    F'0'                TCB ADDRESS
         DC    X'10000000'         NEXT DEB ADDRESS
         DC    X'60000000'         OLD DATASET
         DC    X'0F001000'         OUTPUT PROCESSING
         DC    X'00'               NUMBER OF DASD EXTENTS
         DC    3X'00'
         DC    X'FF000000'         PRIORITY
         DC    X'0F'               THIS IS A DEB
         DC    AL3(0)              DCB ADDRESS
         DC    X'04'               DASD DEB
         DC    AL3(0)
LENDCBDB EQU   *-DCBDEB
         EJECT
WORK     DSECT
WORKLEN  DS    H                   LENGTH OF WORK AREA
DEQCNT   DS    H                   NUMBER OF DEQ/ENQ'S ON CATALOG
TCBADDR  DS    F                   TCB ADDRESS
R7SAVE   DS    F                   REG 7 SAVE AREA
R8SAVE   DS    F                   REG 8 SAVE AREA
R10SAVE  DS    F                   REG 10 SAVE AREA
R11SAVE  DS    F                   REG 11 SAVE AREA
TIMEIN   DS    D                   TIME OF ENTRY
CCA      DS    F                   CATALOG COMMUNICATIONS AREA ADDRESS
CCASAVE  DS    F                   ADDRESS OF CURRENT SAVE AREA IN CCA
MYECB    DS    F
CCW      CCW   X'23',SECTOR,X'60',1  SET SECTOR
SEARCH   CCW   X'31',0,X'40',5     SEARCH FOR R0
TIC      CCW   X'08',0,0,0         TIC*-8
ERASECKD CCW   X'11',0,X'60',0     ERASE
         CCW   X'03',0,X'20',5     NO-OP
MYIOB    DS    0F
FL1      DS    C
FL2      DS    C
SENSE    DS    H
ECBA     DS    F
CSW      DS    2F
CCWA     DS    F
DCBA     DS    F
RESTR    DS    F
INC      DS    F
MYSEEK   DS    2F
*
SDATA    DS    0D
CCHH     DS    XL4   SAME AS IOBCCHH
         DC    X'0100'   R=1, KL=0
         DC    AL2(0)
         DC    C'SCRIBBLE'
```

```
*
SAVE      DS    12F
GTF       GTRACE MF=L
OUTDCB    DS    0F                          DCB FOR DATA BEING ERASED
          DS    12X'00'
DCBDVTBL  DC    F'0'                        ADDR OF ENTRY IN I/O DEV CHAR TAB
          DC    X'00'
DCBDEVT   DC    X'00'
          DC    2X'00'
          DC    F'1'
          DC    H'0'
          DC    X'4000'                     PS
          DC    F'1'
          DC    X'06000001'
          DC    X'C0000000'
          DC    H'0'
          DC    BL2'1101000000001000'
DCBDEBAD  DC    A(0)
          DC    X'9200'
          DC    BL2'1101000000001000'
          DC    5F'0'
OUTIOVEC  DS    0H                          DEB PREFIX
          DC    A(APPEND)
          DC    A(APPEND)
          DC    A(APPEND)
          DC    A(APPEND)
          DC    A(APPEND)
          DC    3F'0'
DEBLEN    DC    X'00000000'                 LENGTH OF DEB IN DOUBLE WORDS
OUTDEB    DS    0F
DEBTCBAD  DC    F'0'                        TCB ADDRESS
DEBDEBB   DC    X'10000000'                 NEXT DEB ADDRESS
          DC    X'60000000'                 OLD DATASET
          DC    X'0F001000'                 OUTPUT PROCESSING
DEBNMEXT  DC    X'00'                       NUMBER OF DASD EXTENTS
          DC    3X'00'
          DC    X'FF000000'                 PRIORITY
DEBPROTG  DC    X'0F'                       THIS IS A DEB
DEBDCBB   DC    AL3(0)                      DCB ADDRESS
          DC    X'04'                       DASD DEB
DEBAPPB   DC    AL3(0)
DEBDVMOD  DC    X'00'
DEBUCBA   DC    X'000000'                   UCB ADDRESS
DEBBINUM  DC    X'0000'                     BIN NUMBER
DEBSTRCC  DC    X'0000'                     START CYLINDER
DEBSTRHH  DC    X'0000'                     START TRACK
DEBENDCC  DC    X'0000'                     END CYLINDER
DEBENDHH  DC    X'0000'                     END TRACK
DEBNMTRK  DC    X'0000'                     NUMBER OF TRACKS
LENDEBEX  EQU   *-DEBDVMOD                  LENGTH OF EXTENT DESCRIPTION
          DC    11F'0'
ENDGET    EQU   *
          SPACE 4
          ORG   OUTDCB
GTREC     DS    0D                          GTF RECORD FORMAT
GTIMEIN   DS    D                           TIME OF ENTRY TO SCRIBBLE
GTIMEOUT  DS    D                           TIME OF EXIT
GTCALLER  DS    C                           SCRIBBLE CALLER CODE (S OR R)
GTCOMP    DS    C                           SCRIBBLE COMPLETION CODE
GTDSN     DS    CL44                        DSNAME
GTVOL     DS    CL6                         VOLUME SERIAL
```

```
GTNERASE DS     CL2                        NUMBER OF TRACKS ERASED
GTNREAD  DS     CL2                        NUMBER OF TRACKS READ
GTNDEQ   DS     CL2                        NUMBER OF DEQ/ENQ'S ON CATALOG
GTNMEXT  DS     CL2                        NUMBER OF EXTENTS RELEASED
GTEXTS   DS     0C                         UP TO 16 10-BYTE EXTENT DESCRIPTS
         SPACE 10
DADSMTBL DSECT                             DADSM EXTENT TABLE
         DS     C
EXTNUM   DS     C                          NUMBER OF EXTENTS IN TABLE
         DS     2C
ENTRIES  DS     16F                        UP TO 16 EXTENTS
         EJECT
         END
```

DOCUMENT CONTROL DATA SHEET

Security classification of this page: UNCLASSIFIED

1 DOCUMENT NUMBERS	2 SECURITY CLASSIFICATION
AR Number: AR-001-985	a. Complete Document: Unclassified
Report Number: ERL-0136-TR	b. Title in Isolation: Unclassified
Other Numbers:	c. Summary in Isolation: Unclassified

3 TITLE

THE ADAPTATION AND INSTALLATION OF THE RESOURCE ACCESS CONTROL FACILITY (RACF)

4 PERSONAL AUTHOR(S):	5 DOCUMENT DATE: April 1980
J.L. Roughan J.C. Gwatking	6 6.1 TOTAL NUMBER OF PAGES 152 6.2 NUMBER OF REFERENCES: 12
7 7.1 CORPORATE AUTHOR(S): Electronics Research Laboratory	8 REFERENCE NUMBERS a. Task: DST 78/044 b. Sponsoring Agency:
7.2 DOCUMENT SERIES AND NUMBER Electronics Research Laboratory 0136-TR	9 COST CODE: 228801/135
10 IMPRINT (Publishing organisation) Defence Research Centre Salisbury	11 COMPUTER PROGRAM(S) (Title(s) and language(s))

12 RELEASE LIMITATIONS (of the document):

Approved for Public Release

12.0	OVERSEAS	NO	P.R. 1	A	B	C	D	E

Security classification of this page: UNCLASSIFIED

Security classification of this page: UNCLASSIFIED

13 ANNOUNCEMENT LIMITATIONS (of the information on these pages):

No limitation.

14 DESCRIPTORS:

a. EJC Thesaurus Terms

Electronic computers
Computer systems programs
Time sharing
Real time operations
Data processing equipment
Revisions
Computer programs
Operating systems (computers)
Source programs

b. Non-Thesaurus Terms

15 COSATI CODES:

0902

16 LIBRARY LOCATION CODES (for libraries listed in the distribution):

17 SUMMARY OR ABSTRACT:
(if this is security classified, the announcement of this report will be similarly classified)

The Resource Access Control Facility (RACF) is a software package designed to control access by users to a computer system and to data stored on the system. This report describes the modifications and additions to the functions of RACF which were made during its installation in the computing centre at the Defence Research Centre. RACF is described in sufficient detail to allow the operation of the modifications to be clearly explained. The report also summarizes the functions and standards of the computing centre and lists the actions taken to accommodate users with non-standard requirements.

Security classification of this page: UNCLASSIFIED